United Business Media

gamedeveloper

THE LEADING GA___ ___Y MAGAZINE

TOP 20 PUBLISHERS

inFAMOUS
POSTMORTEM

"I WON'T READ YOUR MANUAL"
THE COMMUNICATION CONUNDRUM

gamedeveloper

COVER ART: SUCKER PUNCH ART TEAM

PUBLISHING PAPAS

WHO'S YOUR DADDY?

PUBLISHERS. ARE THEY A NECESSARY EVIL?
Developers seem to portray them that way at times, and even the "necessary" part goes away in the indie and online spheres, where a developer can self-release. But evil? I'm not sure.

It's often been said that publishers are only out to make a buck, and the larger they get, the more that can be true. Take, for instance, this quote from an interview I did with Sierra stalwart Mark Hood about his time at Vivendi in the early 2000s: "It basically became sitting down on a panel with eight people, probably three of whom were from the game industry, and the other five were either from a cosmetics company or hair color or water and power company, and they would be approving our games.

"It was like the same questions would come up every time. 'Well, how is this like DIABLO? Tell me how this is like DIABLO.' 'Well, it's not like DIABLO. It's not at all like DIABLO. It's completely different.' 'Oh, well, no. You need to give us a game like DIABLO.'"

The situation has hopefully changed since the Activision merger, but in that scenario, the game is seen in terms of numbers. How much will this make us? The larger a company, the more likely it is that your executives will think this way, whether they came from another game company, or a restaurant chain.

BURY ME WITH MY MONEY
» Someone has to think about the money, and I'm sure you don't want it to be you, who'd rather just get on making a good game. The trouble comes when the money and the creativity appear to be at odds. I'm optimistic, and feel there are ways that the money issues and creativity can fall in line to create something excellent that also makes its money. Somebody greenlit HALO, and CALL OF DUTY, and RESIDENT EVIL 4's three restarts.

Developers and publishers often have a curious relationship. The best analogy I can think of is that of parent and child. The publisher/parent thinks it knows best because it's been there before (shipped more games), and because "it's my house (money), so you'll live by my rules."

The developer/child is rebellious, and thinks it has all the answers. In many ways, it does know more than the parent, and is closer to what's innovative, but maybe hasn't figured out how to hone that energy yet. I could take this analogy further, with talk of advice/feedback loops and misunderstandings, but ultimately, publishers have the money and the marketing, while developers have the creative spirit and know-how.

BECAUSE I SAID SO
» What makes a good publisher then? It seems to vary based on your market. In the case of the iPhone, I've heard developers say that having a publisher is largely useful for marketing. Some might say they take a good game and promote it—Others might say they take a game that would've sold anyway, and exploit it. All depends on how your deal went, I suppose!

For MMOs, a publisher is most likely to be the one serving your game, taking care of customer service, to some extent, and marketing. In general, a third-party publisher isn't going to do much to your game aside from localize it.

It gets more complicated in the console arena, of course, and that's where the back-and-forth parental relationship can come into play. Ultimately, a publisher is only as good as its employees. Some of external producers at the publisher can actually really help focus your work. In this month's postmortem, Sucker Punch mentions that marketing helped them trim the fat.

Publishers sometimes do know where the money is, and money allows you to make more games. What's unfortunate is when they can't see past GTA and GUITAR HERO to see an actual new idea, forgetting that GTA and GUITAR HERO were, at one time, new ideas, or at least clever new amalgams of old ones.

I do think that publishers can definitely help make a game better. On top of marketing and feedback, publishers often also offer external QA, take care of any legal issues that may come up, and pay the bills. But that's only if they're willing to take a little risk, and actually trust the developers they're working with. Incidentally, since both companies should really be doing some proper due diligence on each other, trusting each other shouldn't be part of the "risk" bit.

IP FREELY
» As a developer, your job becomes knowing how to give publishers what they want (more guns!), while also making the game you want (time travel!). As publishers, the risk assessment work should mostly be done at the top end. After that, there needs to be a lot of monitoring, after all they should get the game they pay for, but also a lot of trust.

If you're trying to make a risky game with new ideas, it's best to wrap the concepts in the familiar. Making new IP is always going to be a battle. But if you stay strong, and both parties really listen to each other, it can be a battle that winds up getting you a better-playing and better-selling game.

—*Brandon Sheffield*

remix albums 2009

As participants in an interactive medium, it is only fitting that video game players take ownership of the games written on their hearts. By appropriating the music of video games, remixers and arrangers transmute their favorite works into interesting, new configurations. And it isn't just bedroom producers who are remixing game soundtracks; big publishers are extending the mix as well. Here frequent Gamasutra contributor Jeriaska takes a look some of the outstanding remix albums of the year.

KIND OF BLOOP
http://kindofbloop.com

Andy Baio risked trampling on sacred ground when he set out to arrange *Kind of Blue* for the Miles Davis album's 50th anniversary. Recruiting aficionados of 8-bit chip tunes—AstOr, Virt, Sergeeo, Shnabubula, and Disasterpeace—the album has sparked a contentious debate among listeners as to whether retro gaming and an improvisatory medium like jazz are natural bedfellows or inherently at odds.

Composers: Miles Davis, Bill Evans, Wynton Kelly, Jimmy Cobb, Paul Chambers, John Coltrane, and Julian "Cannonball" Adderley.
Arrangers: AstOr, Virt, Sergeeo, Shnabubula, and Disasterpeace

SCREAM NO HITO, VOLUME 2
www.sweeprecord.com/?p=286

To be fully appreciated, Super Sweep's latest remix album must be supplemented by a series of YouTube videos. They depict the cloaked, mask-wearing Scream villain performing arrangements of classic game themes, including CASTLEVANIA and AFTER BURNER, on the violin. Volume Two caters to those truly in the know by remixing "Morning Music," the tune vintage Konami arcade cabinets played while warming up to an operational 50 ℃.

Composers: Miki Higashino, Masahiko Takaki, Hisayoshi Ogura, and Hiro
Arrangers: Shinji Hosoe, Yousuke Yasui, Takayuki Aihara, Ayako Saso, Norihiro Furukawa, and Scream no Hito

PIANO COLLECTIONS KINGDOM HEARTS
www.square-enix.co.jp/music/sem/page/piano_kh

Yoko Shimomura's first high concept cross-pollination of huge franchises came with her Super Nintendo score to SUPER MARIO RPG. While frenetic energy has been a cornerstone of her compositions from the get-go, this year's collection of piano arrangements of the Disney / FINAL FANTASY mashup also takes moments to linger on more somber and meditative notes in "Dearly Beloved" and "The Other Promise."

Composer: Yoko Shimomura
Arrangers: Sachiko Miyano and Natsumi Kameoka

METROID METAL, VARIA SUITE
http://metroidmetal.bandcamp.com

Beginning as a modest single-track remix by solo musician Grant Henry, *Metroid Metal* has since evolved into a sweeping treatment of the Nintendo series' audio across multiple consoles. *Varia Suite* debuts together with the band's show-stopping performance at the Penny Arcade Expo, and for many the live renditions of "Norfair," "Brinstar," and "Maridia" have proven as infectious as the PAX flu.

Composers: Hirokazu "Hip" Tanaka, Ryoji Yoshitomi, Kenji Yamamoto, and Minako Hamano
Arranger: Stemage (Grant Henry)

MONSTER HUNTER 5TH ANNIVERSARY ORCHESTRAL CONCERT
www.capcom.co.jp/monsterhunter/anniversary/orchestra.html

In Japan the PSP is occasionally referred to as the "MONSTER HUNTER machine" because for some it's their only game for the console. Masato Kouda's music, which mixes regional instruments and classical orchestration, is a factor in the series' massive popularity. The fifth anniversary orchestral concert organized by Harmonics International includes selections from every numbered game installment including the recent TRI, performed by the Tokyo Philharmonic Orchestra.

Composers: Masato Kouda, Tetsuya Shibata, Yuko Komiyama, and Akihiko Narita
Arrangers: Shiro Hamaguchi, Yasunori Iwasaki, Kazuhiko Sawaguchi, Hayato Matsuo, and Akifumi Tada

KAZE ATSUME NO SALMO
http://kou-ogata.net/kaze.html

ZUN's TOUHOU shooters have sparked a renaissance in remix albums by authorizing the sale of derivative works. Previously fan arranger Kou Ogata had taken the latent Celtic influences of FINAL FANTASY and CHRONO TRIGGER and made them explicit by treating them to live performances on Irish and Scottish instruments. His Comic Market 76 release retains the lush acoustic vibe of his previous recordings and stands out amidst a veritable sea of TOUHOU fanfare.

Composer: ZUN
Arranger: Kou Ogata

DODONPACHI DAI-OU-JOU ARRANGE ALBUM
http://cave.shop-pro.jp

The twitch reflex gameplay of Cave's bullet hell shooters has yet to find a solid market outside of Japan, so the arranged soundtracks of Manabu Namiki's DEATHSMILES and KETSUI games have flown under the radar in the West. The most recent remix album for DODONPACHI is exhilarating.

Composer: Manabu Namiki
Arrangers: Yasuhiko Fukuda, Yasuhisa Watanabe, Kohta Takahashi, Ryu Umemoto, Motoaki Furukawa, TECHNOuchi, Kinuyo Yamashita, Hiroki Kikuta, Motoi Sakuraba, Kota Hoshino, Akari Kaida, and Tomoko Sasaki

8-BIT DEMOCRACY
http://lidbjork.homelinux.org/nintendo

Swedish rocker Mutherpluckin' B has self-published two albums conceptually melding melodies of the NES era with '70s funk. Imagine Link trailing Ganon through Shaft's New York back alleys and you've halfway envisioned the soundtracks. *8 Bit Democracy* takes more of a classic Led Zeppelin approach to MEGA MAN and TMNT stage themes. The Socialist Edition is free online, while the Capitalist version ships with a trippy take on the *Godzilla* end theme and more.

Composers: Hiroki Isogai, Yasuaki Fujita, et al.
Arranger: Mutherpluckin' B

MEGA RAN 9
http://random.bandcamp.com/album/mega-ran-9

Arizona-based rapper Random redefined the scope of game remixes in 2007 with the debut of *Mega Ran*, a collection of hip-hop soliloquies by the Blue Bomber and his metallic foes. *Mega Ran 9* continues the rapper's fruitful collaboration with musicians DN3, Samik, and Storyville, adding back-story to the boss battles and turning this year's retro Capcom sidescroller into a parable of pathos and conflicted characters.

Composers: Ippo Yamada, Ryo Kawakami, Yu Shimoda, Hiroki Isogai, Ogeretsu Kun, Manami Matsumae, and Yoshihiro Sakaguchi
Arrangers: Random, Mutherpluckin' B, and Shadix

SANCTUARY / THE STAR ONIONS
www.square-enix.co.jp/music/sem/page/onion_2

Low-key instruments such as the flute, tin whistle, mandolin and accordion make for relaxing riffs on such FINAL FANTASY XI themes as "Gustaberg" and "Distant Worlds." Thoughtfully arranged by experienced musicians of RPG scores, The Star Onions' second arrange album can be downloaded for $10 on iTunes.

Composers: Nobuo Uematsu, Naoshi Mizuta, and Kumi Tanioka
Arrangers: Masato Kouda, Naoshi Mizuta, Takahito Eguchi, and Masakazu Sugimori

HOW PENNY ARCADE EXPO DOES IT DIFFERENTLY

THE PENNY ARCADE EXPO, WHICH DEBUTED IN 2004, WAS CONSTRUCTED OUT OF THE DESIRE OF THE *PENNY Arcade* web comic creators, Jerry Holkins and Mike Krahulik, to create a genuine grassroots event made for their peer group. Fast-forward to 2009, and this year's PAX practically took over downtown Seattle. Everyone, from taxi drivers to TSA officials at the airport, seemed to know about "that convention," and seemed pretty happy that the geeks were in town. A mass of attendees (60,750 according to the shows organizers) swarmed the Seattle Convention Center to geek out on panels from game makers like BioWare and nerd icons like Wil Wheaton, indulge in community-based video game and tabletop game playing, and walk a show floor packed with the high profile games. As a first-time Penny Arcade Expo attendee, and someone who's been to most of the other big consumer and trade video game shows worldwide, from E3 through GDC to GamesCom and beyond, I've tried to distill down what's different about the show to any others I've encountered.

EVERYONE AT PENNY ARCADE EXPO IS A GAMER. It's a subtly different demographic from other shows. For example, at Comic-Con, the fan base is split between comic fans, TV fans, movie fans, and other franchise aficionados. Game Developers Conference is developer-focused, and E3 has a lot of businessmen. Because PAX is a relatively inexpensive consumer-specific game expo— complete with cosplayers and happily rabid fans—almost nobody who attends the expo is in it for the money.

PAX IS A SMOOTH MELDING OF THE VIDEO GAME AND TABLETOP GAME WORLDS. One of the key things about Penny Arcade guys is that they're as geeked out by social board, card, and D&D-like games as they are video games. So, rather than paper game events like GenCon trying to incorporate video games in a potentially unwieldy manner, Penny Arcade Expo is a natural blending of the two. Having $11,000

deluxe wood game tables showcased alongside XBLA downloadable games is a pretty neat concept.

THE PAX PANEL LINE-UP MIXES UP GRASSROOTS AND BIG BUDGET ARTFULLY. It's great that PAX has panels and showcases throughout the three-day conference, and a lot of work has gone into arranging these sensibly. Having content other than the show floor is a major boon to attendees. In addition, much of the line-up feels uniquely PAX-like, particularly the community-specific panels from those who hang out in the fan spaces of game culture, like Mega64, Player vs. Player, or The Guild. But there are also mega-panels for games like HALO 3: ODST to keep things interesting. Some of the panels are a bit unfocused and the "takeaway" may be weak, but this isn't GDC where you have to learn things and tell your boss—it's a spare-time fan convention.

IT'S NOT E3-EXPENSIVE FOR COMPANIES TO PARTICIPATE AT PAX. Having been to both E3 in Los Angeles and GamesCom in Cologne in the last few months, the slightly reduced scale (lower ceilings, less cavernous expo space) of PAX means that exhibitors don't have to spend millions on their booths. The gap between large and small exhibitors is smaller (a good egalitarian trend), and the insane multi-level E3 mega-displays are nowhere to be seen.

PENNY ARCADE EXPO'S HOURS ARE GEEK-FRIENDLY. If there's one thing we know about game fans, it's that they don't exactly follow a 9am–5pm schedule. It's great to see common areas at PAX open practically around the clock, and evening concerts and panels well designed to eat up the night hours with (optional!) fun.

PAX IS A UNIQUE SHOWCASE FOR INDIE CREATORS. PAX allows independent creators from multiple industries to co-exhibit to consumers easily. So,

there are folks like graphic novel creators Oni Press, the *Red Vs. Blue* machinima creators at Rooster Teeth, indie game outfits like Twisted Pixel and Klei, and paper-game folks like Privateer Press all showing alongside each other.

AT PAX, YOU CAN FEEL THE LOVE. If you consider last year's PAX keynote from BioShock's Ken Levine and this year's from Monkey Island and DEATHSPANK creator Ron Gilbert, you'll see a common theme. It's not about how to make games or even how game developers should get inspired (as in Game Developers Conference keynotes). It's about how games make you feel. It's about the emotional response that you, I, and every gamer has to the artform. And that's a wonderful thing, and a hallmark of the entire Penny Arcade Expo experience.

—Simon Carless

Cocos2d for iPhone

COCOS2D FOR IPHONE (WWW.COCOS2D-IPHONE.ORG) is a free, open-source framework for building 2D iPhone games from Ricardo Quesada and his team at Sapus Media. So far it has been used as the foundation for over one hundred iPhone games ranging from STICK WARS to FISHING FRENZY.

The core of what Cocos2d offers is a rendering and image loading solution that uses OpenGLES—a flexible and powerful method for getting whatever graphics and animations you want onto the iPhone fast. The framework also comes with a built-in scene state manager, Z-sorting with layer support, and heaps of nifty features such as scene transitions, text rendering support, tilemaps, and more.

Cocos2d is open source, which means that you can load up the engine and change anything you want. More importantly, you can delve into the murky depths of the engine code and examine exactly how a specific feature works. Looking through the Cocos2d codebase, it's obvious that the people behind it know how to program concisely and intelligently—pretty much everything that I've looked at has made sense at first glance, and in most cases the logic works in the way I'd want it to if I had written it myself.

Cocos2d isn't perfect—the lack of documentation makes the first few steps difficult, although there is a great whitepaper from Monocle Studios that walks you through the first steps of getting something on the screen and there are also some invaluable resources

provided by other Cocos2d users. Due to the framework being perpetually improved and added to—often by grateful iPhone developers—you'll occasionally come across components that are not quite finished, and you may need to make a few modifications to your own code when a new update comes through.

—Jason Bakker, http://dfgame.wordpress.com

Monocle Studios whitepaper: http://monoclestudios.com/cocos2d_whitepaper.html

A primer on Cocos2d's key classes: http://lethain.com/entry/2008/oct/03/notes-on-cocos2d-iphone-development

Making cocos2d a shared library in XCode: www.clintharris.net/2009/iphone-app-shared-libraries

Cocos2d hints and tips from the forums: www.cocos2d-iphone.org/forum/topic/737

Reliable Pathfinding Technology.
Havok AI.

P Physics **a** Animation **b** Behavior **c** Cloth **d** Destruction **ai** AI

havok™
www.havok.com
Turning Creative Aspirations into Technical Realities ™

wrap it up

using SWIG to connect with scripting languages

By Juan Manuel Alvarez

IT HAS BEEN A LONG TIME SINCE GAMES WERE CONSTRUCTED EXCLUSIVELY WITH C/C++ CODE.
When choosing a game engine, productivity and creativity are as important as performance—and C++ alone does not exactly excel there.

A few years ago, several new languages emerged, and quickly became standards in game development. Scripting languages such as Python and LUA are now used by game and level designers to develop gameplay and set game rules, because these languages have simpler syntax, and can be modified without touching the core code. Mono and .NET also have a large number of followers when it comes to creating game editors, since they are very productive and powerful when developing user interfaces.

The problem is these languages need to access the algorithms and data structures created in the core C/C++ game engine, which means those interfaces must be transparent to the friendlier languages. This seems an impossible task when undertaken by hand, so it is normally done using a specialized tool for each target language, like ToLua, Boost::Python, or Managed C++. It creates a big problem when more than one new language has to be used, since different tools have to be learned and wrapper interfaces have to be created and maintained for each tool.

This article focuses on the use of SWIG (Simplified Wrapper and Interface Generator) for automatically wrapping C/C++ interfaces to multiple languages (we will use LUA and C# as examples), removing the aforementioned problems and leaving the door open to integrate more languages in the future.

WHAT IS SWIG?

>> The official definition of SWIG is: "A software development tool for building scripting language interfaces to C and C++ programs." Released in 1996, it provides support for more than 15 languages—including LUA, Python, C#, Java, and PHP—and for the following C/C++ features: pre-processing, all ANSI C/C++ datatypes, functions, variables, constants, enums, pointers, classes, single and multiple inheritance, overloaded functions and methods, overloaded operators,

wrap it up

C++ templates (including member templates, specialization, and partial specialization), namespaces, variable length arguments, exceptions, smart pointers and standard library datatypes.

SWIG works by taking C/C++-like declarations created by the user in an interface file and creating the wrappers needed to access those declarations from other languages.

STARTING WITH SWIG

>> Let's start using SWIG with a simple example. Suppose we have a C++ header file `eng_entity.h` with a class declarated on it (Listing 1).

```
namespace eng {
class Entity {
public:
    void update ();
};
}
```

LISTING 1 eng_entity.h

The simplest way to wrap it to LUA with SWIG is to create an interface file like the one in Listing 2 and compile it with SWIG using the following command line: `swig.exe -c++ -lua -o eng_entity_lua.cpp eng_entity.i`.

```
%module eng {
#include " eng_entity.h"
%}
%include " eng_entity.h"
```

LISTING 2 eng_entity.i

This will make SWIG parse the header file and create a file named `eng_entity_lua.cpp` that will contain all the code that wraps the file content to LUA.

The same can be done with minimal effort to wrap the same code to C# with the following command line: `swig.exe -c++ -csharp -o eng_entity_csharp.cpp--dllimport eng_core.dll eng_entity.i`.

This will not only generate the `eng_entity_csharp.cpp` file, but also an `Entity.cs` file to be used in the C# project. It is very common to make an assembly (managed DLL) with all the wrapped classes, and use them in different projects, such as different tools or editors.

USING THE CLASS IN THE TARGET LANGUAGE

>> Each language has its own needs in terms of initialization and interface registration. In the case of LUA, we have to register the wrapped module into the `lua_state`. Fortunately, SWIG automatically generates a method of doing this. In Listing 3 we can see a simple example of how to register the wrapped functions and classes into LUA.

In Listing 4 you'll find an example of how to use the generated class inside a LUA script. It is important to note that SWIG makes use of the LUA metatable feature, so it will only generate code compatible with versions 5.0 and later.

```
#include "lua.h"
#include "lualib.h"
#include "lauxlib.h"

// declare the function SWIG generates for us
extern int luaopen_eng(lua_State* L);
int main(int argc, char* argv[])
{
    lua_State* pkState = lua_open();
    // load basic libs
```

```
    luaopen_base(pkState);

    // register the wrapped interfaces
    luaopen_eng(pkState);

    // ... load and execute a script
}
```

LISTING 3 C++ code to load the LUA wrappers.

```
e = eng.Entity()
e:update()
```

LISTING 4 LUA script using our wrapped class.

The case of C# is more difficult, since SWIG uses P/Invoke and not Managed C++ to wrap C/C++ to C#. To do this, it generates a `.cpp` file specified in the `-o` parameter that has to be compiled into the DLL specified in the `-dllimport` parameter, in this case `eng_entity_csharp.cpp` and `eng_core.dll` respectively.

CLASSES, PROPERTIES AND MEMBER FUNCTIONS

>> As we have seen in our first example, SWIG is capable of wrapping properties and member functions for C++ classes as well as structs, built-in data types, pointers, references, and arrays in a very easy way. Just use the `%include` directive mentioned in the previous section and all the wrappers will be generated. In the case of C#, SWIG will generate common classes, but in the case of LUA it will use user data.

But if we have more complicated interfaces that make use of templates or output parameters—or if we want to take advantage of target language features—we will need to explicitly tell SWIG about them.

In the following sections we will describe how to use SWIG to generate code using this features, in an easy and non-invasive way.

INHERITANCE

>> SWIG can take advantage of inheritance both in C# and LUA. In the case of C#, it will generate common classes and will use all the object-oriented programming features normally (aside from multiple inheritance, since that is not supported by C#).

In the case of LUA, where OOP is not supported, classes will be wrapped into user data and multiple inheritances will be correctly wrapped. Base-type resolution and polymorphism are also supported. For example, if you have a `Character` class that derives from `Entity`, and if you have a function in C++ that receives an `Entity*`, passing a `Character*` as a parameter will be correctly handled. Also, the overridden methods in `Character` will be called even if they are passed as `Entity`.

ATTRIBUTES

>> One of the best things about SWIG is that it lets us take advantage of target language features. In C# one of these features is properties. These are perfect for editors, since they bind automatically to the well-known PropertyGrid control found in WinForms.

So if we have accessor methods in a `Character` class, like the ones in Listing 5, and we want to wrap them into a property called `Energy`, all we have to do is use the `%attribute` directive as shown in Listing 6.

```
float getEnergy () const;
void setEnergy (float fEnergy);
```

LISTING 5 Part of the Character class declaration located in eng_enemy.h

```
%module eng {
#include "eng_character.h"
%}
%attribute(eng::Character, Energy, getEnergy, setEnergy);
```

LISTING 6 eng_character.i

In C# you can use the `%csattribute` directive to attach C# attributes to wrapped methods and variables like this: `%csattributes eng::Character::Update() "[ThreadSafe(true)]`. Using this we can customize the way properties bind into the `PropertyGrid` control.

In LUA, the SWIG attributes will turn into data attributes, and they must be used as shown in Listing 7.

```
c = eng.Character()
c.Energy = 5
print(c.Energy)
```

LISTING 7 Using attributes in LUA.

TEMPLATES
>> In order to wrap a template you need to give SWIG information about a particular template instantiation and an alias to be used as a typename, since most languages do not accept the characters `<` and `>` as part of a typename. In SWIG this is done using the `%template` directive.

As an example, let's suppose we have a template class `List` and we want to wrap two types of lists in SWIG: `List<int>` and `List<float>`.

Listing 8 shows how to achieve this.

```
%template(IntList) List<int>;
%template(FloatList) List<float>;
```

LISTING 8 Wrapping template classes.

You can also use this mechanism in template functions. For example: Listing 9 shows how to wrap the following C++ template function: `template<class T> T max(T a, T b)`.

```
%template(maxInt) max<int>;
%template(maxFloat) max<float>;
```

LISTING 9 Wrapping template functions.

STANDARD LIBRARY
>> SWIG provides full support for the most common Standard Library data structures, which are: `std::deque`, `std::list`, `std::map`, `std::pair`, `std::set`, `std::string`, `std::vector` and `std::exceptions`. All we need to do is to include the proper `.i` file (for example `%include "std_string.i"`).

In the case of containers, you have to declare them with the `%template` directive after including the proper `.i` file, as seen in the previous session. For example: `%template(IntVector) std::vector<int>;`.

For C#, SWIG will generate the `IntVector.cs` file, which will contain a class of the same name that derives from `System.Collections`.

wrap it up

`Generic.IEnumerable`, and a class `IntVectorEnumerator` that derives from `System.Collections.Generic.IEnumerator`.

In the case of LUA the wrapped vector is more limited. One of its limitations is that all access is made by value by default. To support references, we must do some macro tricks and use some obscure SWIG features.

Of course, not only standard types have to be used. We can also declare containers of our own types and SWIG will take care of the rest. For example: `%template(EntityVector) std::vector<Entity*>;`.

MODIFYING GENERATED CODE

>> When wrapping to a target language, we must follow different coding conventions. For example, we may be writing C++ member functions in lower-case when all the C# programmers expect them in upper-case.

To fix this, we have to make use of the `%rename` directive, as shown in Listing 10, that lets us tell SWIG the target name we want to generate for a specific type or declaration.

```
// rename a class
%rename(MyCharacter) eng::Character;

// rename a member function to upper-case
%rename(Update) eng::Entity::update;
```

LISTING 10 Examples using %rename.

On the other hand, if we want to tell SWIG to ignore certain declarations, we have to make use of the `%ignore` directive as shown in Listing 11.

```
// ignore a class
%ignore eng::Character;

// ignore a member function
// with a certain declararion
%rename eng::Entity::update(int);

// ignore all overloaded member functions
%rename eng::Entity::update;
```

LISTING 11 Examples using %ignore.

This is commonly used to disambiguate, since target languages do not always have a match to all C++ data types.

For example, if we have two overloaded functions like: `void myFunction(int)` and `void myFunction(short)` and we are trying to wrap them to LUA, SWIG will not be able to disambiguate them, so we will need to ignore one to avoid SWIG throwing us a warning.

PREPROCESSING

>> SWIG is well aware of preprocessor macros and fully supports the use of `#if`, `#ifdef`, `#ifndef`, `#else`, and `#endif`. It also defines the macros `SWIGCSHARP` and `SWIGLUA` so we can use specific code to wrap for each target language.

Additionally, SWIG provides an enhanced macro capability with the `%define` and `%enddef` directives to define large macros of code. One of the advantages is that it is not necessary to terminate each line with the annoying continuation character "\"—well known to C++ programmers.

```
%define DECLARE_CONTAINERS(prefix, T)
    %template(prefix ## Vector) std::vector<T>;
    %template(prefix ## List) std::list<T>;
    %template(prefix ## Queue) std::queue<T>;
%enddef
```

LISTING 12 Declaring enhanced SWIG macros.

```
DECLARE_CONTAINERS(Int, int);
DECLARE_CONTAINERS(Double, double);
DECLARE_CONTAINERS(String, std::string);
DECLARE_CONTAINERS(Entity, Entity);
```

LISTING 13 Using enhanced SWIG macros.

ORGANIZING LARGE PROJECTS

>> When wrapping a big library or engine, it would be impossible to wrap everything in a single `.i` file for various reasons: a `big.cpp` file would be regenerated and recompiled even at the smallest change in the interface file, and it would be very difficult to keep track of all the interfaces generated in a single big file, not to mention almost impossible to maintain.

For this, SWIG provides us two directives: `%include` and `%import`. The first one includes the file and generates wrappers for the declarations it finds on it. The second one only collects type information, but does not generate any wrappers.

So, the better way to organize the `.i` files would be to match them with the C++ header files from the library. This way, when changing an interface in `eng_entity.h`, the corresponding `.i` file (`eng_entity.i`) will be very easy to find. Finally, we would need to match the `%include` and `%import` directives with the `#include` directives or the forward declarations in the header files, so SWIG can find the right declarations.

CONCLUSION

>> And this is just the beginning. SWIG is a very big tool which offers you the ability to do a lot more than we could cover in this article— abstract C++ classes extended in the target language, documentation generation, smart pointers, output function parameters, exceptions, type conversion, and the list goes on and on. Now that you have an introduction to SWIG, you can start reading its extensive manual in order to take advantage of all the possibilities it offers.

On the other hand, be patient. As I have said before, it is a big tool and it takes time to make certain things work, and wrap an entire existing library as a game engine. But once you get it working, you will immediately see the benefits of having only one tool and one wrapper file to generate wrappers to multiple languages. Happy wrapping! 🎮

resources

The SWIG official page is www.swig.org. Here you will find the download links and the manuals.

If you plan to start using SWIG, subscribing to the SWIG Users List is a must. Not only are there a lot of helpful people there, but it also serves as a great knowledge base. https://lists.sourceforge.net/lists/listinfo/swig-user.

You can also find a Custom Build Rule for Visual Studio 2005/2008 to wrap `.i` files to C# in http://naicigam.wordpress.com. This way you can easily integrate SWIG into your Visual Studio projects, without messing with command lines and file dependencies.

JUAN MANUEL ALVAREZ is a 3D software engineer and co-founder of Aquadize Studios. He is working on tools and middleware for game development as well as teaching graphics programming. He is also a developer on the Fuzion Media Asset Manager project. Email him at jmalvarez@gdmag.com.

EVEN IF YOUR COMPANY BUDGET EQUALS YOUR CREDIT CARD LIMIT,

your game can be Unreal.

GDC 10

Game Developers Conference®
March 9-13, 2010 | Moscone Center, San Francisco
www.GDConf.com

Learn. Network. Inspire.

PRESS **A** TO JUMP
teaching players to play

EVERYONE AND HIS BROTHER HAS GREAT IDEAS. THE CHALLENGE FOR a talented designer is executing on the good ones. And the hardest part of that is getting the player, who is years and miles away from the initial development process, to understand. And players aren't that bright. Not as smart as us game designers, certainly!

Communication is hard because players are not here to learn; they're here to play. But if they don't learn, they will never know how to play. Sometimes education works best with subtle methods that players barely even notice, but sometimes it requires a large blunt instrument. In either case, the game must make education as seamless as possible if it is going to stick. Too many games rush the education process, which leaves players either clueless about features that would enhance their enjoyment, or annoyed at repetitive and unhelpful information being shoved down their throats. Don't make that game. Think your learning curve through. Use the right method at the right time. Only you can make learning fun.

INFORMATION TECHNOLOGY
» There are three main types of information that games need to communicate to players.

» What can the player do? What are the controls and abilities?
» Why should he do it? What are the goals and rewards?
» How can the player best use these powers and achieve these goals? What are the strategies?

This article focuses on the first of these three pillars; the controls and abilities. This is what the player needs to learn first, which means bolder (though still subtle at times) communication methods are required, compared to the other two pillars. Over the years, game developers have developed a variety of methods to teach controls, some fairly obvious and some less so. Let's start with the obvious ones.

PROMPT

>> Presenting information carefully makes a big difference in the player's enjoyment and learning. Different ways of presenting the same message have varying effects on the player's experience. Choose carefully. One of the core questions to ask of any control information is whether to interrupt the player. If you stop the game's action and put up some text and/or voiceover, it ensures that the player will at least notice it. But that interrupts the flow of gameplay, which can jar players out of the immersion the game has worked so hard to set up.

Most games these days focus on non-interrupting text that the player can react to at his own pace. This has the added advantage of leaving the prompt visible while the player performs the action, so he doesn't have to remember anything between reading and acting. This is especially common in games that are primarily for the core console audience, such as UNCHARTED: DRAKE'S FORTUNE or CALL OF DUTY 4: MODERN WARFARE. More casual games are more of a mixed bag. Games like SUPER MARIO GALAXY or DE BLOB, both of which present their story through interrupting boxes, often do the same for control information, leading to lots of interruptions during story and tutorial moments.

Once a popup has served its purpose, remove it to avoid any risk of annoyance. Players generally don't like being told something they already know. Similarly, with information that is based on a very specific location or object, there's no reason to present the information until the player is ready to interact with it. We used this form of prompt when I was working on CONAN at

THQ—objects such as health vases, or characters such as maidens with a regular gameplay role weren't explained until the player got close to the first one, so if the player found one early or missed the one we expected him to find, he still got the prompt at the right time.

In addition to tailoring the list of popups, presentation matters. Players will detect any hint of annoyance in a popup and react violently to it. ZELDA: OCARINA OF TIME is a wonderful game scarred by a goal reminder hint system that is remembered to this day as "that damn fairy." It followed many of the rules I suggest, but failed on presentation. A high, chirpy voice and over-zealous reminder tone led to an unpleasant nagging experience in an otherwise well-polished game. Remove all traces of condescension and forced cuteness from these texts. Simpler is better.

MINIMIZE

>> One of the most basic mistakes with control information is giving too much of it. The easiest way to avoid annoying a player with information is to cut it in the first place.

Overdoing control information can be a side-effect of the unique way designers see the world. For a designer, the game is a set of connected systems, each with a plethora of critical moving parts that connect in fascinating ways. If you take one out, the rest fall apart or don't make sense. But to a player, new controls can be a new problem to solve, a new skill to learn. They don't think of the connections when they're first experiencing the controls; they just want to make it work.

Information overload is a sin that goes beyond being annoying; it can become counter-productive. Giving players a rapid series of consecutive controls to learn will not help the lessons stick. It will generally make the player dump the information he's already learned and stop paying attention to any future information. The next-gen BIONIC COMMANDO is guilty of this, giving not only too much information in a boxed-in tutorial, but also prompting the player to learn skills the character won't learn in the game for

some time. Each control needs time to practice, learn, master, and store away. KATAMARI DAMACY (PS2) had a great little tutorial sequence, but made the mistake of introducing all the controls at once, separate from the game's goals and world. I went many hours into the game before I remembered the flip and charge controls, despite their clear introduction in the opening sequence.

When a game introduces an innovative new feature, it is usually worth a lengthy control introduction. But simple concepts that players already understand don't need this much time and effort. In some cases, just indicating where to find the control is enough, and players can extrapolate from experience. For example, FALLOUT 3 mostly uses standard shooter conventions, but then adds the VATS concept for paused aiming. This is reflected in the training— the game quickly works through most controls such as jumping and shooting with small text snippets that don't interrupt the action, but lingers over VATS, pausing the game with a full-screen popup with detailed descriptions.

Outside of the innovative new feature how does the designer know what features deserve more hoopla? The only real way to find out what players need is to test it on players. This is a great place for usability testing. Putting representatives of the sample audience in front of early builds of the game will show you where people really struggle. And it won't be where you expect. When working on DRAWN TO LIFE: SPONGEBOB edition, we put the game in front of kids at THQ's usability lab, and found that even the youngest had no trouble with basic movement commands, so we took out those prompts. And we found that despite clear text prompts, some kids were baffled when we required them to use the butt-stomp, so we re-emphasized that prompt.

INTEGRATE

>> Basic information can get annoying. Once it's been narrowed down to the smallest meaningful set of prompts, it is important to make the prompts as fun as possible. Giving the prompts depth with the character and voice of a GLaDoS (PORTAL) or King of All Cosmos (KATAMARI DAMACY) makes the prompts interesting rather than tedious. It requires a bit more work on the writing side, as care is required to ensure that the story information doesn't obscure the control information, but it can be done well. Hire a professional writer if you need to—anything to avoid the aforementioned ZELDA issue.

This is something that games have gotten pretty good at, especially narrative-driven games. Games like BIOSHOCK and FALLOUT 3 introduce concepts during sequences with meaningful narrative value, furthering the story while furthering player controls. Each piece of control information is phrased in the voice of a key character who will have continuing narrative value throughout the game—FALLOUT 3 with the character's dad and a few others, BIOSHOCK with

Commonly understood movement controls are only briefly touched on in FALLOUT 3's tutorial but new systems such as VATS are explained in detail.

Atlas. If the control information is fun to read or hear, people don't mind it.

GATE

>> Telling the player how to do something is rarely enough. For important concepts that players are going to need to learn, it's important to not just teach, but also to test. Immediately after telling the player how to do something, set up a small challenge that requires use of that skill. Set it up so that there is no way around it other than proving basic competency with the new control. Leave the command prompt up until the player actually does it (or at least leave the player with a simple way to call the information back up). There is really no substitute for a clear gating puzzle to ensure that players actually pay attention—but the gate has to be integrated and fun. The player shouldn't feel like they're literally trapped in a box.

Ideally, any gating puzzle will have a few smaller challenges just afterward to reiterate the idea. Repeating the idea helps secure it in the player's mind. If the idea is worth interrupting gameplay with a gate, it's worth cementing in the player's mind with a few follow up challenges. These don't have to be as involved as the first, and maybe shouldn't even be required to complete, but should give the player some fun ways to play with his new abilities. Encouraging experimentation with ample rewards is a great way to teach. Look at games like SUPER MARIO GALAXY, where the first big new gameplay control (spin) is introduced with a big description followed by some crystals that can only be broken with that move, followed by a world with lots of things that break this way. This is another reason that teaching too many concepts is a bad idea—to do it right, each one requires considerable support within the level designs.

EMPHASIZE

>> The above guidelines are primarily intended for the big concepts that require heavy-handed communications methods. But that's not the only avenue for education. There are methods that don't have the same success rate, but are great for supporting information, and reiteration of key ideas.

Use as many methods as you can to emphasize additional information. Loading screens are a great time to present controller maps or gameplay tips. Printed manuals are a great reference tool for players who care. Putting that information inside a pause menu or other in-game resource is even better. Don't assume anyone will ever use it, but for those who are curious it can be a great boon. Integrating tips into the game world can be a great way to make them even more palatable, similar to the story integration suggestions listed above. Consulting with a mentor character mid-game or accessing a useful database can make the player feel better about needing help and make it more fun.
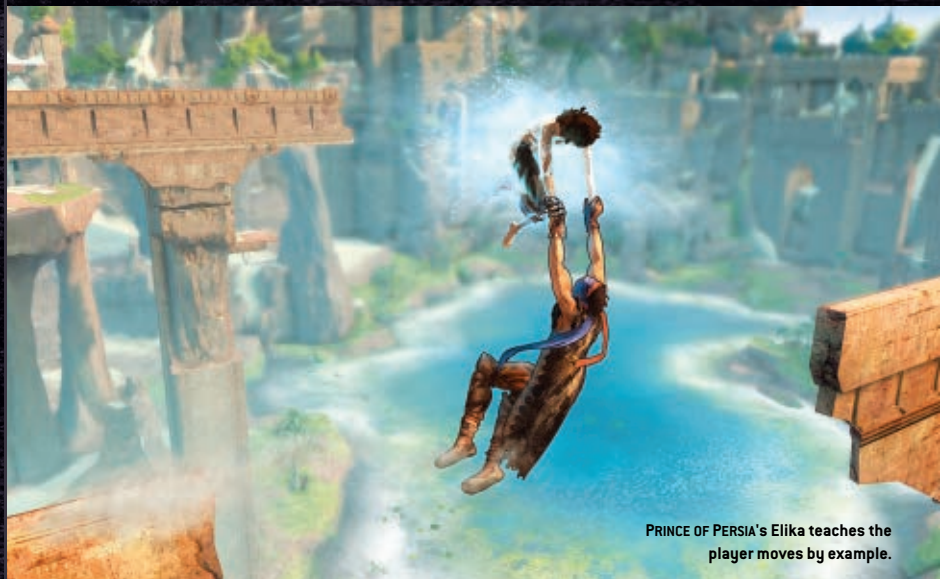
PRINCE OF PERSIA's Elika teaches the player moves by example.

ADVANCED TECHNIQUES

A lot of the above was perhaps obvious groundwork for the experienced designer, who likely wants some more advanced methods. Designers who can get the player to learn without even noticing it have truly mastered the craft. Let's explore some of what they do to make this happen.

SHOW ME: THE MISSOURI PRINCIPLE

>> When you want the player to do something, sometimes it's best to simply show him. When someone else does something interesting, players will want to emulate it. And seeing an ability in action is the best education on how to use it—the situation and result are right there with no confusion. Games like KIRBY take this to its logical extreme where you can directly steal enemy powers, but any weapon-based game that lets you take flamethrowers off of dead Nazis has some of the same appeal.

This is one of the secret driving principles of multiplayer games—anything awesome quickly propagates through the community as players see other people do it and try to mimic it themselves.

Friendly allied characters are also an easy way to do this. For example, the next-gen PRINCE OF PERSIA opens with a sequence in which you chase another character through a chasm, copying her moves. Seeing her do it gives you the basic idea much better than any text or even graphic could.

You can also use boss battles to introduce powerful new abilities, forcing the player to learn how to master the ability in the hands of an enemy before he can use it himself.

THE ASSUMPTION PRINCIPLE

>> If you put something in the game that players are familiar with, they will expect certain things from it. This is the interactive equivalent of Hitchcock's Gun—if the player sees something, especially something cool, they're going to want to see it do its thing. This can be used productively to communicate controls, but more often it's a landmine that threatens to give the player ideas that the design never intended.

This can be filtered through the style and character—if you're a cyborg super-spy, players might not expect to be able to go down the slide when they pass through a playground. But some will try, and they'll like the game better if it works, because putting in a slide suggests slide controls. If sliding is something you do in active gameplay contexts, this will become more of a reasonable expectation than an extra.

Assumptions can also help with communication. Framing systems in the context of things the player is familiar with makes them easier to understand. MAFIA WARS generally makes sense within player's expectations of the Mafia—they want to own weapons, do violent tasks, and give things to the Godfather. The Godfather feature is a good example of using this principle to the game's advantage, as it frames a feature that the designer wanted (microtransactions for increased power) in a way that fits player expectations (trading favors with the boss), making the feature feel natural and obvious. A slightly different phrasing on that feature would make it much harder for players to understand.

Character designs can be built around assumptions. If the goal is for the player to want to run up to enemies and maul them bloody, a chainsaw placed prominently in shots of the main

STREET FIGHTER IV features characters whose combo moves can be discovered through button mashing.

character gets this idea across nicely. And if the player is meant to burn down certain enemies, making those enemies out of wood avoids lots of other communication expectations. To get the most from this, it has to go both ways. Designers may need to tweak features to make them better fit the real world in places where the two don't align. The communications gains may be worth a gameplay compromise.

The inverse of this can tend to cause trouble for designers. If you put something in the game and it doesn't do what the player wants, the player will get frustrated. If there is a toilet, the player is going to want to try to flush it, so put a sound there. This often requires massive backstory planning to avoid constant frustration—there's a reason so many games are set in post-apocalyptic futures where most machines don't work (except ones that make games more convenient, like lights, doors, and guns).

THE STREET FIGHTER PRINCIPLE

>> Make it so the features you want players to see are easy to stumble upon through other actions. I've named this feature after STREET FIGHTER, which is famous for the ease with which players can button mash their way into combos that they have no understanding of. It may sound like an awkward design, but in reality it works very well because players will use the basic moves (easily understood by the one-to-one correspondence with physical buttons) in ways that tend to also trigger advanced combos. And once the player has stumbled upon a move, they know that it exists and can start extrapolating backward to learn how they did it. Button mashing leads the player to better understanding of the system.

This principle can apply to abilities outside of button mashing as well. When the player can interact with a variety of different objects, they will often tend to stumble upon different abilities just by walking around. One of the reasons so many games reward players with shiny things for exploring their physical space is that it encourages the player to stumble upon features that they might not otherwise notice. Similarly, breakable objects don't usually need a lot of direct teaching since players are likely to accidentally smash a few things while doing their normal attack abilities. In these cases, as long as there's a clear and obvious result that the player can tie back to the action, it explains itself.

THE "YOU'RE SOAKING IN IT!" PRINCIPLE

>> One good way to teach is to force the player to do something even if that wasn't quite what he meant. It's similar to the STREET FIGHTER principle, but requires slightly different setups. A good example of this can be found in RATCHET AND CLANK FUTURE (PS3). Grinding on rails is a movement mode that is introduced early in the game. Many games with grind rails require the player to jump onto them to start the grind. But in RATCHET, it shoves you onto the rail as soon as you get near one, using a region that looks similar to the already-introduced jump pads. So players who are just wandering

around exploring (gameplay encouraged by generous reward crates) are likely to stumble upon the trigger zone even if they don't know what it's going to do. And once the player is on the rail, the gameplay starts no matter what the player does.

This same logic can be applied to a variety of different systems. Many games force combat onto the player whether they want to or not. In the first HALO, the player is put in that Warthog no matter what he does. Consider ways to elegantly shove players to where they need to be. Sink or swim is an underrated learning method.

THE BUSH DOCTRINE: THE PRE-EMPTIVE STRIKE

>> The goal of removing popups as soon as possible can be done pre-emptively. If a player exhibits a certain ability before the game prompts him to do it, popups related to that control should never play. If the player picks up the controller and starts running around and shooting without any help, don't tell him how to run around and shoot. It doesn't work when the topic is more abstract, like teaching about goals or rewards. But it's surprising how many top tier games leave this info to appear to experts. For example, SUPER MARIO GALAXY teaches you how to jump no matter how many times you've already jumped. Sure, it's a key concept, but a few lines of code are worth it to remove a minor annoyance for expert players.

This concept can be taken even further, using the player's initial actions to extrapolate what sort of tutorial would be best for them. CALL OF DUTY 4: MODERN WARFARE does a little of this, using the player's performance in the training course to suggest a difficulty level (that the player can override). It's rare to see, but as games reach for wider audiences with more disparity in play ability, it should become more common. For example, if a player starts up a game for the first time and immediately starts using genre-specific controls, you can guess that they're probably comfortable with those controls, so you can set those as the default controls and avoid any related messages.

THE PRETTY PICTURES PRINCIPLE:

>> Using more than just text can be a great way to express complex information to players. Images convey certain types of information better than any thousand words. TMNT shows acrobatic moves with a simple concept sketch that shows how the character will move. DE BLOB shows both character motion and player motion at the same time, which is especially helpful on the Wii where there is no standard language to explain Wii motions to casual players. ROCK BAND doesn't provide much control information at all, but one of the few prompts it does use is a simple visual to remind drummers to use the foot petal if they go a while without pressing it.

THE ROPES

>> These are just a few of the sneaky ways that games make people "accidentally" learn things without really noticing it. These methods don't preclude the basic initial lessons though. In all cases, keep it short, simple, and integrated, and avoid anything anywhere close to annoying.

And this is just the start. Once the player has learned the what of playing, the same techniques can apply to the why and how. The other pillars, mentioned at the article's start, can be a bit harder to teach, as they're more abstract and require players to understand larger systems. Gating based on understanding of a system can be hard, but when it works, it helps players not just play the game, but fully understand the design. Encouraging players to dig deeper and think harder about games can only bring about good things in the future.

TOM SMITH *wrote this during the brief period between his creative manager gig at THQ and his shiny new senior producer gig at Disney mobile. He's worked on oodles of games, ranging from HUNTER and CONAN, to STITCH and SPONGEBOB. Email him at* **tsmith@gdmag.com.**

Unreal Technology News
by Mark Rein, Epic Games, Inc.

Canadian-born Mark Rein is vice president and co-founder of Epic Games based in Cary, North Carolina.

Epic's Unreal Engine 3 won Game Developer magazine's Best Engine Front Line Award for three consecutive years, and it is also the current Hall of Fame inductee.

Epic's internally developed titles include the 2006 Game of the Year "Gears of War" for Xbox 360 and PC; "Unreal Tournament 3" for PC, PlayStation 3 and Xbox 360; and "Gears of War 2" for Xbox 360.

Upcoming Epic Attended Events:

KGC
Seoul, Korea
October 7-9, 2009

Dubai World Game Expo
Dubai, United Arab Emirates
October 27-29, 2009

DICE Summit
Las Vegas, NV
February 17-19, 2010

GDC 2010
San Francisco, CA
March 9-13, 2010

Please email:
mrein@epicgames.com
for appointments.

UNREAL ENGINE 3-POWERED ANIMATION KIT, MUVIZU, LAUNCHES PUBLIC ALPHA

Glasgow-based DA Group recently released the public alpha of Muvizu 3D, a new animation package powered by Unreal Engine 3 that lets users of all skill sets create 3D animations with no experience required.

The Muvizu application, which includes all the assets and tools needed to create 3D movies, is available for free download from www.muvizu.com, a portal that enables users to collaborate on film, audio and drama projects with easy-to-use tools and viewable results in real time.

"We chose Unreal Engine 3 because it is proven technology with an excellent support network," said Vince Ryan, managing director, consumer unit, DA Group.

Muvizu 3D scene in playback mode

"Furthermore, the engine accepts assets created in Maya, and this will lower the barrier to entry for our users when everyone will eventually be able to submit their own creations for others to use within the Muvizu community."

The philosophy of Muvizu is for content creators to put less energy into animating characters and more effort into directing them onscreen. The goal is to foster collaborative filmmaking by uniting people with contrasting skills through Muvizu's virtual workspace and storyboarding capabilities.

During Muvizu's development, DA Group relied heavily on Unreal Engine 3 for rendering, physics and especially scripting.

"The power of the scripting system allows code to be developed much more rapidly than if the entire engine were in C++," said DA Group's lead developer, Robert McMillan.

McMillan explained how his team came to work with Epic and ramp up to hit tight production deadlines by leveraging the Unreal Developer Network.

"In the early days, we had been in touch with Epic about using Unreal for another project that never went ahead. We'd had discussions with Mark Rein, who even came to visit our office. We didn't know what to expect, but the help and assistance was really good. This set the tone for our expectations this time around, and it was no different. In addition, Epic were really flexible with us while the terms of the license were negotiated."

McMillan continued, "In terms of the software, we got more than we expected. We found through UDN that people have been building add-ons for the engine for quite some time. Technology integrations for features such as animation blending systems and UI tools are available, so UDN saved us from having to do this type of integration work ourselves.

"There is definitely a feeling of community, and that community is dedicated, patient and helpful beyond the call of duty," said McMillan. "Without UDN, we would probably have had to delay our launch for many, many months."

"We had expected a professional relationship with Epic and not much more at first," said Ryan. "The reality, however, was far better. We were encouraged and supported all the way by Epic. Nothing was too much trouble, and UDN seems almost to harbor a collegiate relationship. Truly remarkable and very, very helpful. It was nice to discover that, despite us being a minnow compared with other licensees, we were still accorded first-rate and friendly service."

"The level of support and information that is available from UDN is incredible - with access to guys at Epic who really know the engine inside out," said McMillan. "I am so glad we went with Epic, and I really don't think we would be in the position we are in right now had we not made the right choice."

Now that Muvizu 3D is available to the public, DA Group is working with its growing user community to bubble the most ingenious projects up to the top for all to enjoy. The studio is launching a competition that will award a high-end hardware prize to the best video created with Muvizu 3D, and the excitement will only grow as more creative types try out the application.

For UE3 licensing inquiries email:
licensing@epicgames.com

For Epic job information visit:
www.epicgames.com/epic_jobs.html

W W W . E P I C G A M E S . C O M

POWERED BY
UNREAL
TECHNOLOGY

T R E V O R   W I L S O N

top20 publishers

THE PAST YEAR OF ECONOMIC DIFFICULTIES HAS INSPIRED NO SMALL AMOUNT of turmoil and reorganization within the game industry. Arcade pioneer Midway finally succumbed to financial problems and had its major assets purchased by Warner Bros. Interactive. Eidos Interactive, long in search of a buyer, finally found one in Japanese RPG-focused publishing house Square Enix. Several publishers cut a number of studios through sale or by closure, and several others added talent and studios continually throughout the year.

New on this year's survey are independent publisher Bethesda (ZeniMax) and Viacom subsidiary MTV Games. Plus, localization publisher Atlus makes a return appearance after being absent in 2008. Atari (AKA Infogrames, which has taken the Atari name to refer to its entire business) also returns to our survey.

As the game landscape continues to shift, a number of non-retail-oriented game publishers have risen to prominence. Official methodologies for tracking these companies are still in their nascent stages, and we were unable to include them in this year's official list. We are confident that this will be possible next year, but in the mean time, we present an independent and entirely subjective top 10 list of download-only publishers (see page 21).

This year's ranking was calculated by considering the number of worldwide releases, average review scores, and revenue for the period reaching from August 2008 until July 2009. We've also factored in the results of a Gamasutra.com survey we conducted which gathered opinions on 30 major publishers, with a spot to write in others. Over 800 industry professionals were asked to first give their opinions on the reputations of each publisher in the survey, or any we had missed. They were also asked for any specific comments they might have on each of the publishers. Finally, specific feedback on publishers in the form of number scores and comments was gathered from respondents who had direct experience with said publishers. Each of these factors was carefully weighted to produce the ranking you see below.

[A full list of statistics, ratings, and the complete survey will be available in the Top 20 Publishers 2009 report from the Game Developer Research division. For more information, check www.gamedevresearch.com.]

20 LUCASARTS

LUCASARTS

YEAR FORMED: 1982 **HEADQUARTERS:** SAN FRANCISCO **STUDIO:** SAN FRANCISCO

LucasArts had a rather tumultuous year, releasing the externally-developed "legacy" and Wii SKUs of INDIANA JONES AND THE STAFF OF KINGS, while holding back its own internally-developed INDIANA JONES titles for the PS3 and Xbox 360. The popular STAR WARS THE FORCE UNLEASHED was released in this time frame, and is largely what carries the company onto our list this year.

Unfortunately, LucasArts' review average suffered by almost 10 percent compared to last year, as the INDIANA JONES games, and its own internally-developed FRACTURE failed to make a significant impression on critics. The reputation comments all echoed a similar sentiment, well summed-up here: "While it's great to see all the classic LucasArts titles coming back, they need some new IPs."

Commenters who had worked with LucasArts had mixed things to say, with some stating that they "like the changes" the company is making, and that it's "trying to do the right thing," but others mentioned that management didn't understand what makes a good game. Looking forward for the company, it seems as though a renewed focus on PC games and classic franchises could be a rejuvenating factor.

19 ATLUS

ATLUS

YEAR FORMED: 1986 **HEADQUARTERS:** TOKYO **STUDIOS:** SHINJUKU, TOKYO, JAPAN

Though revenues fell slightly for this small publisher, additional releases and better review scores helped give Atlus a spot in our list. Atlus has expanded its lineup consistently, year over year, earning a successful business and a faithful fanbase by mining a niche consisting of RPGs and more experimental titles.

Atlus has an interesting business model, by which the company has measured expectations for each title it releases, and ships only limited quantities, expecting to sell out above break-even. This allows the company to release more games at lower quantities. Still, the company does have its hits. December's PERSONA 4, developed by the Japanese parent company's development house, sold over a half-million copies worldwide and was a critical darling, gaining a review average of over ninety percent. Plus, the RPG gained praise for its inclusion of a character whose sexuality is ambiguous and ultimately determined by the player's choices.

As a localization house primarily, Atlus is in a unique position on our list—most of the games the company released in the U.S. were released in Japan and Europe by other publishers.

18 ATARI

YEAR FORMED: 1983 (INFOGRAMES) **HEADQUARTERS:** LYON, FRANCE **STUDIOS:** CRYPTIC STUDIOS (LOS GATOS, CA); EDEN GAMES (LYON, FRANCE)

Atari had little commercial success to brag about until this summer's GHOSTBUSTERS, which has gone on to sell over a million copies across all platforms, while also achieving reasonable approval from critics (although we should note that the game was published by Sony on certain systems in Europe). Reviewers also liked the Vin Diesel vehicle CHRONICLES OF RIDDICK: ASSAULT ON DARK ATHENA and handheld versions of the indie platformer N+.

Atari has made efforts throughout the year to bring on fresh talent and new IPs, and the acquisition of CHAMPIONS ONLINE and STAR TREK ONLINE developer Cryptic Studios should take the publisher further toward the diversity it will need to make up for the loss of the lucrative *Dragon Ball Z* license.

Atari's main win this year was staying solvent, something that was not guaranteed over the last several years. Former Sony Worldwide Studios boss Phil Harrison also left the company's top management echelons in the year discussed here, and the company is powering ahead without a notable figurehead, perhaps more stealthily as a result.

17 DISNEY INTERACTIVE STUDIOS

YEAR FORMED: 1988 **HEADQUARTERS:** GLENDALE, CA **STUDIOS:** AVALANCHE SOFTWARE (SALT LAKE CITY); FALL LINE STUDIO (SALT LAKE CITY); GAMESTAR (CHINA); PROPAGANDA GAMES (VANCOUVER, BRITISH COLUMBIA); BLACK ROCK STUDIO (BRIGHTON, UK); JUNCTION POINT STUDIOS INC. (AUSTIN, TX)

Though Disney's video game publishing unit released 13 more games during this survey period than it did last year, the company posted lower overall revenues. Notable successes this time around included DISNEY SING IT, HIGH SCHOOL MUSICAL 3, and CLUB PENGUIN: ELITE PENGUIN FORCE.

Disney is set to release a new Spectrobes title in the coming months, part of a Japanese-American collaboration that has resulted in over a million units sold. Disney's work with original IPs like PURE and SPLIT/SECOND impressed several survey commenters.

Looking ahead, Disney's recent acquisition of Marvel Entertainment opens a host of licensing and development opportunities in the game space, which could help propel both companies further ahead—so long as the relationship remains true to both brands.

top20 publishers

16 WARNER BROS. INTERACTIVE ENTERTAINMENT

YEAR FORMED: 1995 **HEADQUARTERS:** BURBANK, CA **STUDIOS:** MONOLITH PRODUCTIONS (KIRKLAND, WA); SNOWBLIND STUDIOS (BOTHELL, WA); SURREAL SOFTWARE (SEATTLE, WA); TT GAMES (BUCKINGHAMSHIRE, ENGLAND); WB GAMES (KIRKLAND, WA); WB GAMES CHICAGO (CHICAGO, IL)

This rapidly-expanding division of media giant Time Warner moves onto our survey thanks to the strong sales of Traveler's Tales' LEGO BATMAN and favorable critical reception for F.E.A.R. 2: PROJECT ORIGIN from the company's Monolith Studios.

WBIE brought a wealth of new talent under its umbrella through the acquisitions of Snowblind Studios, Surreal Software, and the bankrupt Midway Games' Chicago studio. In the process of the Midway deal, Warner Bros. Interactive also gained the potentially lucrative MORTAL KOMBAT license, which will likely be featured prominently in future years. (Would anyone be surprised by a LEGO MORTAL KOMBAT?)

With the success of LEGO INDIANA JONES and BATMAN, and WBIE's newly-acquired control over future *Lord of the Rings* games, the publisher is "definitely up and coming" and poised for additional growth. Many of those surveyed commented to this effect.

15 NAMCO BANDAI GAMES

YEAR FORMED: 1950 (BANDAI); 1955 (NAMCO) **HEADQUARTERS:** TOKYO **STUDIOS:** BANPRESTO (TOKYO); BEC CO., LTD. (TOKYO); NAMCO TALES STUDIO, LTD. (TOKYO); SURGE (SANTA CLARA, CA); YOKOHAMA; TOKYO

2009 saw the creation of Surge, a new publishing label aimed at the Western market, which is where Namco Bandai noted it sees most of its current and future growth. Western sales pushed brawler SOUL CALIBUR IV to over 2.3 million units, though Surge's first title, AFRO SAMURAI, has posted less impressive sales of 420,000 as of this writing. Neither did the Samuel L. Jackson vehicle help Namco Bandai's Metacritic average, which fell several points from last year.

The company's financials reported falling revenues and mounting losses for the year, but Namco Bandai continues to expand via the acquisition of the Japanese D3 Publisher, which in turn owns Vicious Cycle and its Vicious Engine. Perhaps most importantly, Namco Bandai has regained the Western rights to future *Dragon Ball*-related titles, an almost guaranteed cash cow.

14 MTV GAMES

YEAR FORMED: 2007 **HEADQUARTERS:** NEW YORK CITY **STUDIOS:** HARMONIX (CAMBRIDGE, MA)

Viacom/MTV Games' ROCK BAND-focused games-publishing unit comes onto our survey for the first time this year. Viacom subsidiary Harmonix has turned ROCK BAND into a consistent performer, with a new sequel and additional content available for the title year-round.

ROCK BAND 2 itself sold 1.7 million copies through the end of 2008, and further retail and downloadable expansions to the series have been similarly successful. Critics were generally lukewarm on series expansions, but RB2 itself hovered around a 91 percent average Metacritic score across all platforms.

Though it was not released in the period considered, THE BEATLES: ROCK BAND has compared favorably against rival GUITAR HERO 5 as far as critics are concerned. As peripheral-bundled software declined this year, next year will be a critical one for all music-oriented franchises.

13 CAPCOM

YEAR FORMED: 1979 **HEADQUARTERS:** OSAKA, JAPAN **STUDIOS:** CAPCOM INTERACTIVE (LOS ANGELES); COSMIC INFINITY (BURLINGTON, ONT.); TEAM 1 (OSAKA); TEAM 2 (OSAKA)

This publisher saw record growth thanks to massive success of several key franchises, both in the West and at home in Japan. RESIDENT EVIL 5 became the second-best-selling title ever for Capcom, and sold over five million copies across two platforms.

The MONSTER HUNTER series continues to be a huge seller in Japan, with the Japanese version of MONSTER HUNTER FREEDOM UNITE moving over 3.5 million copies as of this writing. STREET FIGHTER IV revived a dormant series to commercial and critical success, and the downloadable BIONIC COMMANDO REARMED proved a runaway hit. Survey commenters were pleased with the way Capcom "actually listens to the Western audiences."

Indeed, Capcom has proved itself one of the only Japanese publishers to understand both domestic and foreign markets to the point that it can excel in both.

download revolution

Digital distribution-focused publishers are on the rise, and here we present to you 10 of the bigger or more interesting players. This is not meant to be an exhaustive list, merely an editor-chosen teaser for the time when we can get the proper data we need to include them in our larger list. The definition of publisher in this space is a bit difficult, with many of the below being developers as well, or "publishing" more in the sense of distributing. With that as a caveat, let's have a look at 10 important digital download-focused publishers, in alphabetical order.

—*Brandon Sheffield*

BLIZZARD: Removing the company from its parent, Blizzard is the company that proved online games could not only work, but allow you to purchase a rocket car and monkey butler. Not that any of the Blizzard execs have any of those. But they could! With over 10 million active subscribers, Blizzard single-handedly proves that with the right product, the subscription model still works. But as the acting juggernaut, it would take a lot to topple WoW. It's a wonder that people keep trying to target the same space directly.

CHILLINGO: Chillingo is a U.K.-based iPhone publisher which tends to release content originating from Europe. IDRACULA, MINIGORE, and ZENBOUND are a few of the company's recent successes, and Chillingo has done a good job of associating itself with high-end titles, so that consumers know to look for the name. Notably, the games Chillingo publishes seem to have helped usher back in an era of Eurogames—games with an actual European character, rather than just getting lumped into the "Western" designation.

GALA-NET: Gala-Net is a Korean free-to-play company with an English language focus through its gPotato portal. As one of the many localization-oriented online publishers, Gala-Net has brought titles such as FLYFF, RAPPELZ, and UPSHIFT STRIKERACER to Western markets. The company is particularly MMORPG-focused, and is nearing 5 million users. gPotato is still relatively small, but is a good example of the free-to-play localization houses that cropped up in North America in the early-mid 2000s. Companies like Outspark, K2 Network, and others share this space.

GAMEHOUSE: Before Facebook apps began to dominate the casual/social gaming mindshare, PC casual portals were kings—and financially, they still may be. GameHouse offers both single-game and subscription purchase plans, the latter of which opens up all games on the portal. As part of the Real networks, which include RealPlayer and associated products, GameHouse is one of the largest players in the area, publishing both its own titles and the games of others—and has recently started branching out into Facebook apps, as that's apparently where the wind is blowing.

NEXON: Nexon was one of the earliest proponents of the free-to-play and microtransactions models, both in the U.S. and its native Korea. A large portion of the online game industry seems to be moving in that direction, but Nexon was out in front from the start. In the U.S. the company also pioneered in-game cash purchasable via cards in retail stores, to the point that now Target and other stores have dedicated racks just for game cash cards. Add to that the company's constant evangelizing of the model to other developers and publishers, and you've got a rather important group of people.

NEWGROUNDS: Though its banner advertising business model is a bit archaic, Newgrounds is the most senior of flash portals and publishers, having been around since the 90s. The company is also very friendly to users and developers, allowing anyone to make money by generating traffic to those ads. Though some may view the financial model as a dinosaur, Newgrounds does get over 500,000 unique views daily, which can actually make the model work for the single-person team. While it won't make any developer a millionaire, it does get the content in front of the right eyes.

NGMOCO: When Neil Young announced that he'd be starting an iPhone-oriented publisher after his departure from EALA, a lot of people thought he was crazy. While the jury is still out on Young's sanity, it's clear that ngmoco is doing well on the new platform, with games like ROLANDO and TOPPLE. On top of that, the publisher has launched its own free community gaming platform, Plus+, which is similar to XBLA or Pogo, and also features games from other publishers. This is reason alone to put ngmoco on our list—the company isn't content to simply release games into a void without a network to support them.

SHANDA: As (arguably) the biggest online game publisher in China, Shanda has a lot of room to grow, with approximately 1.3 billion people populating the country. With $494.3 million in revenue last year, recent entry into the stock market (through a spinoff Shanda Games IPO), and an upcoming WORLD OF WARCRAFT competitor approved by the ruling communist party, it seems as though Shanda is continuing to move up in the world. The government ties are scary, but that's what it takes to operate over there, and you can't deny almost 10 million active paying accounts, either.

VALVE: Through Steam, Valve has done an excellent job of championing digital download as a distribution model for more "traditional" paid-for games on the PC, and has also done a lot of useful experimentation with prices, release dates, discounts, and so forth. Steam has proven to be widely accepted, both by consumers and developers releasing their content on the service. Valve has inspired others to enter the space as well, such as Direct2Drive.

ZYNGA: Zynga is probably the biggest player in the Facebook and social gaming space, and an expanding space it is! MAFIA WARS and FARMVILLE have taken social networks by storm, and some games link to iPhone versions as well. Social games were once the mistrusted newcomer into the game space, with many questioning how they would even make money (the answer became, of course, microtransactions). With Zynga's revenue reportedly on the ups and millions of users of its games, this segment of the industry doesn't look so silly anymore.

top20 publishers

12 SEGA

YEAR FORMED: 1952 (SEGA); 1975 (SAMMY) **HEADQUARTERS:** TOKYO **STUDIOS:** THE CREATIVE ASSEMBLY (WEST SUSSEX, U.K., FORTITUDE VALLEY, AUSTRALIA); PROPE (TOKYO); SECRET LEVEL (SAN FRANCISCO); SEGA SHANGHAI STUDIOS (SHANGHAI); SEGA STUDIOS (TOKYO); SEGA STUDIOS USA (SAN FRANCISCO); SPORTS INTERACTIVE (LONDON)

Last winter's Sonic Unleashed was the biggest contributor to Sega's revenues in the past year, with 2.5 million units sold. Ongoing revenue from the multimillion-selling Mario & Sonic at the Olympic Games was an important presence, as well as sales of Virtua Tennis 2009 and the Japanese release of Ryu ga Gotoku 3.

The publisher did see better critical reception for its titles overall, in a period that saw fewer licensed titles, and began to show some positive results from its publishing deal with PlatinumGames (MadWorld specifically). Sales for the period were down as a whole though, resulting in lower revenues and higher losses.

Sega has been aiming for greater collaboration with Western talent recently, and as part of the fruit of this labor, Empire: Total War sold over 800,000 copies and managed a 90 percent review average. The company's focus on core-oriented Wii titles has yet to yield financial success, but certainly has pleased those fans that purchased them.

11 KONAMI

YEAR FORMED: 1973 **HEADQUARTERS:** TOKYO **STUDIOS:** KONAMI DIGITAL ENTERTAINMENT (LOS ANGELES); HUDSON SOFT (TOKYO, SAPPORO, REDWOOD CITY, CA); KONAMI COMPUTER ENTERTAINMENT (TOKYO); KONAMI SOFTWARE SHANGHAI; KOJIMA PRODUCTIONS

Konami posted higher overall revenues for the past year, thanks to ongoing sales of Metal Gear Solid 4 (which have now passed the 4 million mark) and continued success of its Pro Evolution Soccer series. In a yearly report, Konami noted that its soccer titles broke a previous record to sell over nine million copies across platforms in the past year, and that unit sales of Konami software have consistently improved year-over-year since 2006.

Metacritic review scores were up slightly as a whole for this publisher, and our survey respondents were slightly more kind to Konami this year than last. The company has struggled in its Western expansion, and the Double Helix-developed Silent Hill Homecoming was not particularly well received in Europe or the U.S., and has just been officially canceled in Japan. The partnership with Climax Entertainment for Silent Hill: Shattered Memories looks to rejuvenate the series.

10 MICROSOFT

YEAR FORMED: 1975 **HEADQUARTERS:** REDMOND, WA **STUDIOS:** 343 INDUSTRIES (REDMOND, WA); BIGPARK (VANCOUVER, BC); LIONHEAD STUDIOS (GUILDFORD, U.K.); MICROSOFT GAME STUDIOS JAPAN (TOKYO); RARE (TWYCROSS, U.K.); TURN 10 (REDMOND, WASH.); WINGNUT INTERACTIVE (WELLINGTON, N.Z.)

Microsoft's game-publishing division tightened its belt a bit this year with the closing of its internal studios Ensemble (Halo Wars) and ACES (Flight Simulator). Microsoft had a relatively slim lineup for the year, though it ended up with a very favorable review average and several major hits.

Gears of War 2 was a top seller in November and went on to move over five million copies as of last May, and last October's Fable II became the biggest-selling RPG on the system with 2.6 million units moved. Several survey commenters had kind words for their experiences with Microsoft and praised its "excellent professionalism."

The pending release of the company's Natal motion peripheral will likely influence the publisher's position next year, as it looks to expand into more casual markets, and (hopefully) leads the way on that platform with entries from Lionhead, Rare, and others in its stable.

09 SQUARE ENIX

YEAR FORMED: ENIX (1975); SQUARE (1983) **HEADQUARTERS:** TOKYO **STUDIOS:** BEAUTIFUL GAME STUDIOS (LONDON); COMMUNITY ENGINE (JAPAN); CRYSTAL DYNAMICS (REDWOOD CITY, CA); EIDOS HUNGARY; EIDOS MONTREAL (CANADA); EIDOS SHANGHAI (CHINA); IO INTERACTIVE (COPENHAGEN, DENMARK); TAITO CORP. (JAPAN); SQUARE ENIX CHINA (BEIJING); TOKYO; SQUARE ENIX CO., LTD, (LOS ANGELES)

Square Enix made strides toward targeting the Western market this year by forming new partnerships with overseas studios Double Helix and Gas Powered Games, plus the surprise purchase of U.K. publisher Eidos (though the results from that arrangement are not reflected in the time period covered). If the high scores from our respondents—the highest in our detailed survey—are any indication, Square Enix has already done well at forming relations with Western talent.

In Japan, the Dragon Quest series continued to bring reliable success, with new title Dragon Quest IX selling over three million copies to date, and the Dragon Quest V remake moving another million, though the series has yet to mirror its domestic successes abroad. Further Kingdom Hearts and Final Fantasy series entries also proved successful.

The Taito subsidiary has been making interesting moves as well. As the once-giant arcade company tries to find a new identity under Square Enix, a slew of interesting downloadable titles have been released, largely surrounding reinventions of the Space Invaders franchise. Taito also manages Office Create's Cooking Mama and spinoff series in Japan, to continued success.

top20 publishers

08 THQ

YEAR FORMED: 1989 **HEADQUARTERS:** AGOURA HILLS, CA **STUDIOS:** BLUE TONGUE ENTERTAINMENT (MELBOURNE); SANDBLAST (SEATTLE, WA); INCINERATOR (CARLSBAD, CA); JUICE GAMES (WARRINGTON, U.K.), KAOS STUDIOS (NEW YORK); MIDWAY STUDIOS SAN DIEGO (CALIFORNIA); RAINBOW STUDIOS (PHOENIX); RELIC ENTERTAINMENT (VANCOUVER); THQ STUDIO AUSTRALIA (SPRING HILL, AUSTRALIA); THQ WIRELESS (CALABASAS HILLS, CA); UNIVERSOMO (TAMPERE, FINLAND); VIGIL GAMES (AUSTIN); VOLITION (CHAMPAIGN, ILL.)

This year brought decreased sales and fewer releases (specifically in the licensed sector) for THQ, triggering a wave of layoffs and cost reductions. The publisher closed or sold many internal studios during the past year, including Heavy Iron, Helixe, Incinerator Studios, and Locomotive Games—many of which were responsible for THQ's multiplatform licensed titles, or were very recently acquired.

But the company was not without significant success—SAINTS ROW 2 sold over 2.8 million, and has led THQ management to focus more on similar large "core titles." A surprise during the year was UFC UNDISPUTED (1.5 million sold), which defied analyst expectations and became a critical and retail achievement by all accounts. The latest in the WWE SMACKDOWN VS. RAW series also performed well.

As THQ recovers from a difficult year, the company has picked up shuttered publisher Midway's San Diego studio, including all assets except for the TNA IMPACT! rights. The big question for the publisher go is how it will fare with a reduced volume of licensed children's content from Nickelodeon and Pixar, traditionally the company's bread and butter.

07 BETHESDA SOFTWORKS (ZENIMAX MEDIA)

YEAR FORMED: 1985 **HEADQUARTERS:** ROCKVILLE, MD **STUDIOS:** ID SOFTWARE (DALLAS); BETHESDA GAME STUDIOS (ROCKVILLE, MD); VIR2L STUDIOS (ROCKVILLE); ZENIMAX ONLINE STUDIOS (HUNT VALLEY, MD)

Bethesda comes onto our survey on the strength of the self-published, multimillion-selling RPG revival FALLOUT 3. The title earned Bethesda the highest review average on our survey and the second-highest reputation score, which responders accompanied with praise for FALLOUT 3's "quality downloadable content" and predictions that the publisher "will become one of the big ones."

Zenimax Media reflected this ambition in the surprise acquisition of longtime independent id Software in June, at which time studio head John Carmack also praised Bethesda's handling of the FALLOUT franchise. Further expansion was seen in Bethesda's plans for a new FALLOUT developed by Obsidian and in plans to publish Rebellion's

ROGUE WARRIOR, Splash Damage's BRINK, and A2M's action title WET.

With the id purchase, Zenimax is now in possession of a licensable game engine, which could potentially affect the company's future financials, and also help propel the company's internal projects.

06 SONY COMPUTER ENTERTAINMENT

YEAR FORMED: 1993 **HEADQUARTERS:** TOKYO **STUDIOS:** BIGBIG STUDIOS (WARWICKSHIRE, U.K.); BEND GAME STUDIO (BEND, OR); BANGALORE, INDIA; CAMBRIDGE, U.K.; CLAP HANZ (YOKOHAMA); CONTRAIL (TOKYO); EVOLUTION STUDIOS (CHESHIRE, U.K.); FOSTER CITY, CALIF.; GUERRILLA GAMES (AMSTERDAM); INCOGNITO ENTERTAINMENT (SALT LAKE CITY); SCE STUDIO LIVERPOOL (LIVERPOOL, U.K.); SCE LONDON STUDIO (LONDON); POLYPHONY DIGITAL (TOKYO); SAN DIEGO; SANTA MONICA; SEOUL; SN SYSTEMS (BRISTOL, U.K.); SONY ONLINE ENTERTAINMENT (AUSTIN, TX; DENVER, CO; LOS ANGELES; SAN DIEGO, CA; SEATTLE; TAIWAN); TOKYO; ZENER WORKS (TOKYO); ZIPPER INTERACTIVE (REDMOND, WA)

Sony's first-party publishing efforts gained the company a consistently high review average this year (in line with last year), led especially by high scores for KILLZONE 2, MOTORSTORM: PACIFIC RIFT, and INFAMOUS. The first two titles also gained commercial success, with KILLZONE 2 selling 1.67 million copies and PACIFIC RIFT crossing the 1 million mark.

Survey scores for the publisher's reputation and publishing practices were favorable across the board, and commenters praised Sony's professionalism and production values. One in particular noted that Sony "give[s] appropriate resources to developers to achieve quality."

On that note, more indie developers are checking out Sony's PlayStation 3 for PSN downloadable games, as the publisher makes a conscious effort to be more open. This, apparently, more than makes up for the generally lower sales on PSN versus Microsoft's competing Xbox Live Arcade. Now that Sony Online Entertainment is fully a part of SCE and the company has cut prices on its flagship hardware, next year is wide open for change.

INFAMOUS

05 TAKE-TWO INTERACTIVE

YEAR FORMED: 1993 **HEADQUARTERS:** NEW YORK **STUDIOS:** 2K CZECH (BRNO, CZECH REP.); 2K BOSTON (QUINCY, MA); 2K AUSTRALIA (CANBERRA, AUSTRALIA); 2K MARIN (NOVATO, CA); CAT DADDY GAMES (BELLEVUE, WA); FIRAXIS GAMES (HUNT VALLEY, MD); KUSH GAMES (CAMARILLO, CA); PAM DEVELOPMENT (PARIS, FRANCE); ROCKSTAR LEEDS (LEEDS, U.K.); ROCKSTAR NORTH (EDINBURGH); ROCKSTAR SAN DIEGO; ROCKSTAR TORONTO; ROCKSTAR VANCOUVER; VISUAL CONCEPTS (NOVATO, CA)

The publisher of the GRAND THEFT AUTO series posted lower revenues and experienced a drop in review scores, though its release lineup was slightly larger compared to last year. Quarterly reports also noted losses throughout the year, but 2008's GTA IV continued to sell well throughout the period considered, and in March Take-Two noted that the game had sold over 13 million copies worldwide.

Other hits during the past year included MAJOR LEAGUE BASEBALL 2K9 and ongoing sales of CARNIVAL GAMES. Some survey responders praised Take-Two's "amazing creative talent" and noted that the publisher's "feedback on quality is incredible," but others expressed concern that the publisher may "have trouble branching out."

The company had made a serious bid for downloadable games, such as GRAND THEFT AUTO IV: THE LOST AND DAMNED and THE BALLAD OF GAY TONY, but in a recent statement said that retail would be the company's main driver for the near future. It may be some time before Take-Two makes another large run at the downloadable space.

GTA IV: THE LOST AND DAMNED

GTA IV: THE BALLAD OF GAY TONY

04 UBISOFT

YEAR FORMED: 1986 **HEADQUARTERS:** MONTREUIL-SOUS-BOIS, FRANCE **STUDIOS:** ACTION PANTS (VANCOUVER, BC); ANNECY, FRANCE; BARCELONA; BLUE BYTE (DUSSELDORF, GERMANY); BUCHAREST; CASABLANCA; CHENGDU, CHINA; DIGITAL KIDS (NAGOYA, OSAKA, JAPAN); HYBRIDE TECHNOLOGIES (MONTREAL); MASSIVE (MALMÖ, SWEDEN); MILAN; MONTPELLIER, FRANCE; MONTREAL; MONTREUIL, FRANCE; PUNE, INDIA; QUEBEC CITY; RED STORM (MORRISVILLE, NC); REFLECTIONS (NEWCASTLE, U.K.); SAO PAULO, BRAZIL; SHANGHAI; SINGAPORE; SOUTHLOGIC (PORTO ALEGE, BRAZIL); TORONTO; KIEV, UKRAINE

Ubisoft saw a slight drop in revenues compared to the previous year thanks to a release schedule with fewer megahit IPs, but the publisher still expanded its presence throughout the year, opening and acquiring development studios all around the world.

SHAUN WHITE SNOWBOARDING and FAR CRY 2 were noted to have sold over three million copies each, and the PRINCE OF PERSIA reboot and RAYMAN RAVING RABBIDS: TV PARTY also crossed the platinum boundary to reach 2.5 million and 1.5 million sales, respectively. Respondents to our survey called Ubisoft a "friendly place to work" and praised its "strong capitalization on the *Tom Clancy* franchise."

This year the company plans to open a new Toronto studio, and has some innovative new ideas for the SPLINTER CELL series, as well as the strong-looking ASSASSIN'S CREED 2, and the film-based AVATAR. The company's continued expansion seems unstoppable for now, as Ubisoft's core and casual titles continue to perform well on all platforms.

RAYMAN RAVING RABBIDS: TV PARTY

FAR CRY 2

top20 publishers

03 ACTIVISION BLIZZARD

YEAR FORMED: 1979 (ACTIVISION); 2000 (VIVENDI GAMES) **HEADQUARTERS:** SANTA MONICA, CA **STUDIOS:** BEENOX (QUEBEC CITY); BIZARRE CREATIONS (LIVERPOOL, ENGLAND); BLIZZARD CONSOLE (ALISO VIEJO, CA); BLIZZARD ENTERTAINMENT (IRVINE, CA); BLIZZARD NORTH (SAN MATEO, CA); BUDCAT CREATIONS (LAS VEGAS); FREESTYLEGAMES (LEARNINGTON, ENGLAND); HIGH MOON STUDIOS (CARLSBAD, CA); INFINITY WARD (ENCINO, CA); LUXOFLUX (SANTA MONICA, CA); NEVERSOFT (ENCINO, WOODLAND HILLS, CA); RADICAL ENTERTAINMENT (VANCOUVER); RAVEN SOFTWARE (MIDDLETON, WI); REDOCTANE (SUNNYVALE, CA); SHABA GAMES (SAN FRANCISCO); TOYS FOR BOB (NOVATO, CA); TREYARCH (SANTA MONICA, CA); VICARIOUS VISIONS (TROY, NY); Z-AXIS (FOSTER CITY, CA)

The combined revenue of Activision and Blizzard produced the third-highest sales figures in our survey, driven by such chart-topping franchises as WORLD OF WARCRAFT, CALL OF DUTY, and GUITAR HERO. WoW's total franchise sales topped 8.5 million units in the past year, and the last two CALL OF DUTY titles, MODERN WARFARE and WORLD AT WAR, have topped 13 million and 11 million sold, respectively.

Last year's merger resulted in a shakeup among the two publishers' many satellite studios, with some falling under the new banner, and some being sold off or closed. Outside of GUITAR HERO, the company has largely existed in the core space. But WORLD OF WARCRAFT keeps the company's offerings diverse, pioneering in an online space that many are still struggling to enter.

Blizzard's delay of STARCRAFT II into 2010 should make next year an interesting one and Battle.net is a particularly important product for Blizzard as it increasingly moves toward digital distribution.

CALL OF DUTY 4: MODERN WARFARE

02 ELECTRONIC ARTS

YEAR FORMED: 1982 **HEADQUARTERS:** REDWOOD CITY, CA **STUDIOS:** CRITERION (GUILDFORD, U.K.); DIGITAL ILLUSIONS CE (STOCKHOLM); EA BYRNEST (MOUNT SINAI, NY); EA CANADA (BURNABY, BRITISH COLUMBIA); EA CHINA (SHANGHAI); EA LOS ANGELES (PLAYA VISTA, CA); EA KOREA (SEOUL, KOREA); EA MOBILE (BUCHAREST, ROMANIA; HYDERABAD, INDIA); EA MOBILE KOREA (SEOUL, KOREA); EA MONTREAL; EA NORTH CAROLINA (MORRISVILLE, NC); EA SINGAPORE; MAXIS (EMERYVILLE, CA); EA PHENOMIC (INGLEHEIM, GER.); EA TIBURON (MAITLAND, FL); EA SALT LAKE (BOUNTIFUL, UT); EA TURKEY (ISTANBUL AND ANKARA, TURKEY); BIOWARE CORP. (EDMONTON, ALBERTA; AUSTIN, TX); PANDEMIC STUDIOS (LOS ANGELES, CA; BRISBANE, AUSTRALIA); VISCERAL GAMES (REDWOOD CITY, CA)

This publishing powerhouse is number two for the third year in a row. EA faced a bit of a rough year in which some key new IPs failed to meet sales expectations, and revenues were below what the company hoped, leading to layoffs in many internal studios.

But THE SIMS series' continued success (THE SIMS 3 topped 3.7 million copies sold) and WARHAMMER ONLINE: AGE OF RECKONING gave EA higher revenues, and attention to quality in EA's key sports series led to higher review scores for the FIFA and NHL series. And once again, EA had more releases than any other publisher by far.

The company's best-selling Wii title, EA SPORTS ACTIVE, was also released during the period. Looking forward, EA is continue to focus on its sports and driving strengths while branching out into other arenas as well. The company hasn't given up on new IP after the critically successful MIRROR'S EDGE and DEAD SPACE, and will be taking both series back to consoles in the near future.

DEAD SPACE

01
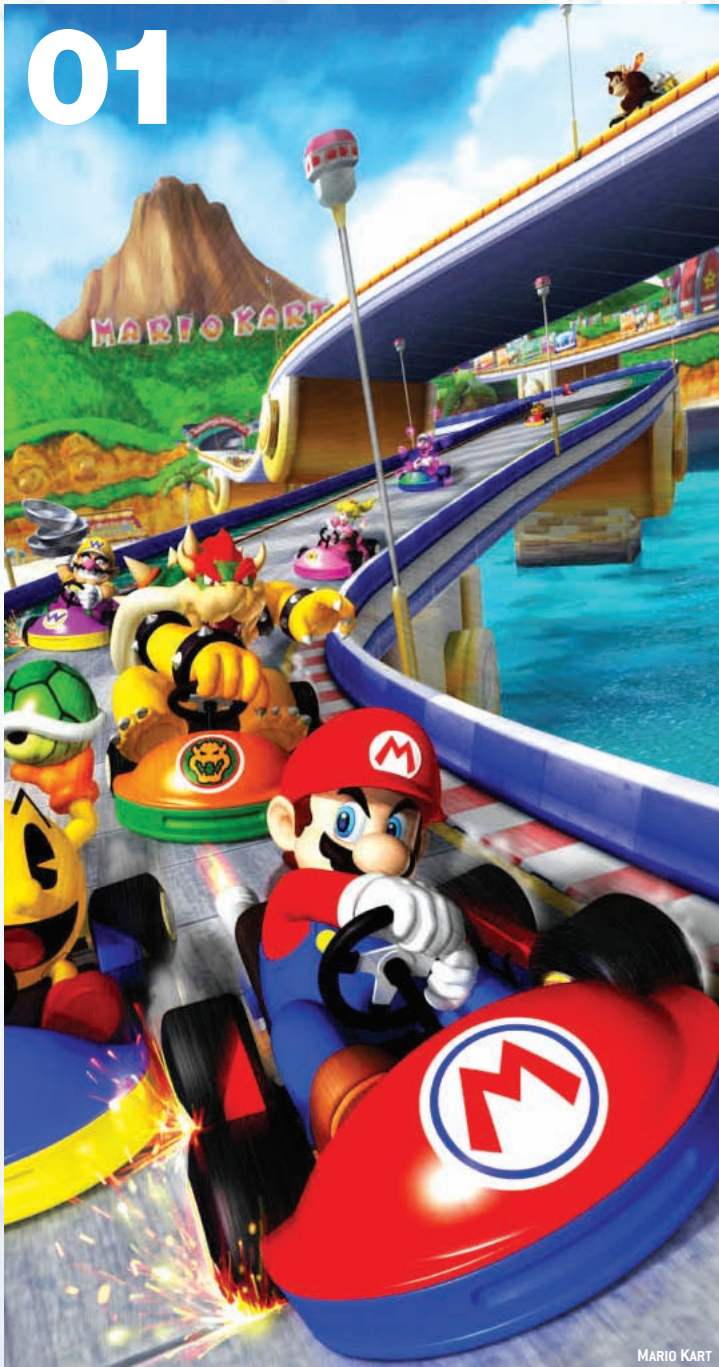
NINTENDO

Nintendo®

YEAR FORMED: 1889 **HEADQUARTERS:** KYOTO, JAPAN
STUDIOS: INTELLIGENT SYSTEMS (KYOTO); NINTENDO
ENTERTAINMENT ANALYSIS AND DEVELOPMENT (TOKYO);
NINTENDO SOFTWARE TECHNOLOGY CORP. (REDMOND,
WA); RETRO STUDIOS (AUSTIN); SYSTEMS RESEARCH &
DEVELOPMENT (KYOTO, OSAKA); BROWNIE BROWN (TOKYO);
NDCUBE (TOKYO); MONOLITH SOFT (TOKYO); PROJECT SORA
(TOKYO)

On top of our survey once again, Kyoto-based
Nintendo continued to increase its already-massive
revenues for the third year in a row, once again
garnering the highest sales on this survey (over $6.5
billion this time around). Multimillion-selling, family-
friendly titles WII FIT, WII PLAY, SUPER SMASH BROS.
BRAWL, and MARIO KART WII continued to top charts into
2009, in spite of several not even releasing during
the period considered.

Plus, the perennially-selling BRAIN AGE series
crossed over 30 million units sold during the year. New
titles also found success, including POKEMON PLATINUM
(over 3.5 million), KIRBY SUPER STAR ULTRA (over 2
million), RHYTHM HEAVEN, and WII SPORTS RESORT.

Nintendo has continued to widen the market
in terms of what the industry considers "console
gamers," and has expanded the gaming landscape
considerably. As a publisher, the company is
meticulous and deliberate in terms of its releases,
having very few duds, if any, while still being willing
to experiment.

Many have criticized Nintendo for existing in
its own world, but this is likely what has made the
company so successful. The company continues to
learn from both its missteps and its successes, and
is certainly evolving in its own direction, with which
the rest are still trying desperately to catch up. 🎮

MARIO KART

WII SPORTS RESORT

top20 publishers

Publisher	RANK 2009	2008	2007	2006	2005	Final Score	Number of Releases	Average Game Review Score	Reputation Ranking	Detailed Survey	Number of Internal Studios
Nintendo	**1**	1	1	2	4	287.82	26	74.86	7.9	8.82	9
Electronic Arts	**2**	2	2	1	1	244.66	146	71.43	7	6.09	20
Activision Blizzard	**3**	3	3	3	2	230.03	124	65.13	7.7	6.56	19
Ubisoft	**4**	4	4	8	6	183.61	98	69.56	7.7	6.27	25
Take-Two	**5**	6	6	5	10	157.18	44	68.57	7.1	6.68	14
Sony Computer Entertainment	**6**	5	8	4	5	149.30	33	74.73	7.3	6.46	21
Bethesda Softworks	**7**	-	-	-	-	146.78	4	91.33	8.1	6.80	4
THQ	**8**	8	5	7	8	146.14	68	69.44	6.5	5.78	13
Square Enix	**9**	10	11	13	16	142.59	19	72.19	7.4	8.83	10
Microsoft	**10**	9	9	6	3	142.53	9	78.33	7.2	6.46	7
Konami	**11**	11	15	9	7	141.12	34	66.92	7	4.67	5
Sega	**12**	7	7	10	9	137.00	35	67.57	6.4	6.10	7
Capcom	**13**	14	14	14	15	135.10	22	71.8	7.1	5.93	4
Namco Bandai	**14**	13	12	11	11/17	124.91	22	61.6	6.3	4.74	5
MTV Games	**15**	-	-	-	-	123.59	12	78	-	-	1
Warner Bros Interactive	**16**	-	-	-	-	122.36	20	68.05	5.9	7.56	6
Disney Interactive	**17**	16	17	18	-	119.78	35	60.92	5.4	6.01	6
Atari	**18**	-	-	19	13	118.44	29	69.24	5.4	4.46	2
Atlus	**19**	-	18	-	-	117.52	23	68	6.1	6.45	1
LucasArts	**20**	17	19	17	20	117.07	14	65.36	6.7	6.15	1

TREVOR WILSON *is a web developer and freelance journalist from Utah. Email him at* **twilson@gdmag.com***.*

Sucker Punch

inFAMOUS

FIRST OF ALL, I'D LIKE TO APOLOGIZE FOR ALL THE EXTRA CAPITAL LETTERS IN INFAMOUS. THE GAME'S ABOUT the ethical choices a superhero would make, and how your choices might result in you becoming famous, or you becoming, well ... sigh. Hey, it could have been worse—we started off spelling it in|famous!

Prior to starting work on INFAMOUS, we'd spent six years producing the three games in the SLY COOPER series. The SLY COOPER games are light-hearted platformers starring the wily master raccoon thief Sly Cooper and his colorful companions. After six years, though, we were a little stale on light-hearted platformers, and the core of the market had drifted away from this sort of game. With a nudge from our publishers at SCEA, we decided to take a swing at building a more mainstream game.

This ended up being more of a challenge than we expected. In retrospect, of course, it should have been obvious. We switched genres. We moved from individual levels to a streaming, open-world city. We switched from family-friendly "toon shading" to a gritty, realistic, decaying cityscape. Our target rating in the US moved up a notch, from E10+ to T, with implied changes to our characters and storytelling. To top it all off, we needed to switch target platforms, from PlayStation 2 to PlayStation 3.

Meeting the challenge took us three and a half years, with much personal and corporate growth for all of us required along the way. It's difficult to single out five things that went right and wrong, in the sea of lessons learned. This post-mortem tries to identify the most interesting and surprising lessons.

inFAMOUS

postmortem
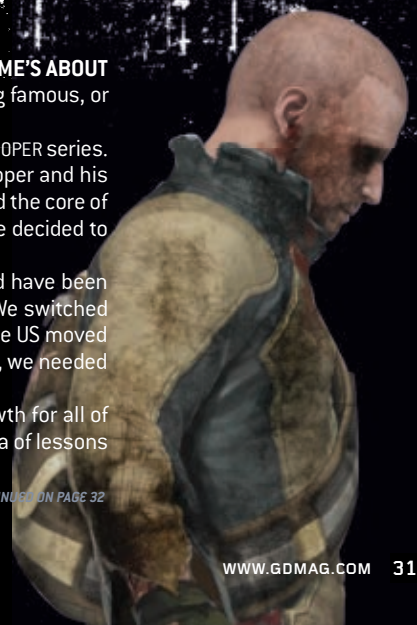
WHAT WENT RIGHT

1} City Construction. From the outset, we knew INFAMOUS would be competing with products built by bigger teams, with bigger budgets. To succeed, we needed to figure out how to generate a large city, with realistic levels of detail, without hiring an army of modelers and texture artists. We decided to focus on building reusable content; rather than modeling and texturing buildings only to use them once, we designed most buildings to be used a handful of times, in different parts of the city.

Furthermore, the buildings all use one of a limited number of footprints—18 in the final game. This isolated the buildings themselves from the layout of the hex-shaped blocks, typically containing 3 to 5 buildings, which form the basic streamable unit of the game. "Hex" layouts can be reused independently of the buildings on them, since buildings sharing the same footprint can easily be swapped. And the buildings themselves can be customized, swapping out the street level storefronts, adding billboards and signage, and so on.

All the stuff on a hex that isn't part of a building—streets and sidewalks, stop lights and street lights, dumpsters, debris and garbage, information for our pathing system, and so on—is placed or modeled once for each hex, with building-footprint-shaped holes left for the buildings themselves. All of this work was reused each time the hex was reused. We had roughly a hundred hexes in the game, but only 27 different hexes, so hexes saw a lot of reuse.

The risk in this sort of reuse is repetition, but the reward is a city 2 to 3 times as big as a city built entirely of custom-fit pieces. We felt the risk was worth the reward, and it paid off for us.

2} Jumping and Climbing. Early on, we decided that parkour-style jumping and climbing would be a core ability for the game's hero. We wanted to integrate the environment into the player's tactical experience, rather than relegate it to a backdrop for the action. We'd had success with this approach in the SLY COOPER games, and thought applying the same techniques to INFAMOUS would be a cakewalk.

We were wrong. It was a rolling disaster, eating up a year and a half of senior engineer time as we iterated through failed approaches. Our goal was to let the player grab and climb anywhere this looked plausible, but we wildly underestimated how many places in a realistic city look climbable. The resulting density of options made figuring out the player's intent from controller input a real challenge.

We call this process of inference "reading the player's mind through his thumbs," and believe it's the key to combining functionality and approachability in a game. Controllers are pretty crummy input devices, really. Translating some imprecise joystick movement and poorly timed button presses into the vast universe of possibilities for what a real person can do requires some serious guesswork. Do a good job, and your game is immersive, with the player focused on the experience, rather than the controller; do a bad job, and the controller is thrown at the screen. It's the difference between the players blaming themselves for a mistake, and blaming the game for the mistake.

Eventually we stumbled upon an approach that worked pretty well, and the resulting climbing and jumping is one of the best parts of the game. In our final jumping mechanic, we start by letting the player air-steer, while we look for climbable points near the player's manual trajectory. If the player steers close enough to a climbable point, we auto-correct the input to steer Cole to land exactly at the climbable point, rather than merely nearby.

If there's any lesson here, it's that sometimes it's worth blowing a ton of effort on a single feature ... but only if you're pretty darned sure that it's going to be a defining feature of the game!

3} Early Collaboration with Marketing. Historically, we've viewed the marketing of our games as a mysterious and completely external process, one in which we played an essentially passive role. We designed and built games; as the game neared completion, the marketing team started figuring out how to sell the thing.

For INFAMOUS, our collaboration with marketing started much earlier, and the SCEA marketing team was much more integrated into the development process. This was massively valuable both for us and for the marketing team. Understanding how marketing viewed the game concept—and how they intended to sell it—made it much easier for us to focus on the right things.

In some ways, marketing acted as a surrogate user for us early on in the development of the product. We would have loved to do focus testing with real users, but the game wasn't polished enough for this. In our experience, focus testing an unpolished game doesn't get you useful information; you just end up hearing about how unpolished the game is.

For example, after a year of concept and initial development, we were still woefully unfocused. One of the key turning points was talking to

We are actively recruiting across all disciplines for the following locations:
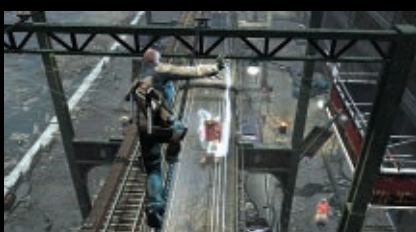
Irvine, California | Austin, Texas | Velizy, France | Cork, Ireland

Seoul, South Korea | Shanghai, China | Taipei, Taiwan

www.blizzard.com/jobs

SCEA marketing about the game, and how they would position it. "It's a game about becoming a superhero," they said. "We're going to market the superpowers."

"Well, yeah," we thought. "Then why are we working on all this other stuff?"

Pretty quickly, we pivoted to a focus on using superpowers as the core of the game experience, and the extraneous stuff drifted away. For example, we'd done man-months of work on prototypes of how the hero might kiss agreeable civilians. This was fun to prototype ... but the input of marketing was crucial to us recognizing it as extraneous fluff.

4 } Schedule Adjustment. Our original schedule showed the development of INFAMOUS taking three years. Two years into the project, it became clear that we were behind schedule. We might have been able to finish a game on the original schedule, but the result would have been a mediocre game.

This realization was pretty painful for us. We'd shipped our last two games within a day or two of their original schedules, and were proud of this record. Somewhat reluctantly, then, we agreed with SCEA that the game's schedule needed to change.

Adding another six months to the schedule turned out to be an excellent decision. We didn't use the extra time as an excuse to add features; we kept cutting extraneous features and narrowing the project's focus. The team stayed focused across the schedule change as well; everyone rallied behind the idea that the extra time provided us the opportunity to do a great game, rather than being forced to scramble getting something out the door.

5 } Prototype, Test, Iterate. One of the first lessons we learned at Sucker Punch was the difficulty of designing a fun video game on paper. We've never had much luck predicting which ideas will end up being fun, and which ones won't. The successes we've had have been the result of trying out lots of ideas, pursuing the ones that had a smattering of fun in them, then testing and iterating until we'd refined the fun into its purest form.

For example, INFAMOUS started with an "ammo" model much like other shooters—defeated enemies dropped ammo for your superpowers. Ammo came in multiple colors, and each superpower used a particular kind of ammo. One kind of ammo needed to be used right away, and timed out if not used quickly enough. Enemies also dropped health packs, which looked like a different color of ammo to the player.

There were good reasons for all of this complexity: we wanted to limit the use of the most powerful hero abilities, we wanted to encourage players to use a broad set of their superpowers, and so on. We'd been uncomfortable with the way the system worked, but felt boxed in by all the constraints we'd placed on the design.

The focus test trumped all of this—players hated the ammo model. They were confused, they didn't like having to conserve ammo, they didn't think picking up glowing balls made them feel like a superhero ... it was a complete train wreck.

The focus test forced us to radically simplify things. After some false starts, we unified ammo and health around the hero's core electrical powers. Drawing electricity from the environment replenishes the hero's store of energy, which is then used up by his powers. Drawing electricity also heals damage to the character. The player ended up with an easy-to-understand model, and a simple rule to follow—when in trouble, look for electricity to drain. At the same time, we gained a powerful way to limit the player's abilities—if an area of the city was blacked out, the player would be relatively weak in that area.

Overall, this change was a home run. Without our core belief in focus testing, we would have a hard time making this large a departure from shooter conventions.

WHAT WENT WRONG

1 } Initial Lack of Vision. For the first year of the project, we really didn't have much of an idea of what kind of game we were building. We were sure about some things—that you played as a superhero, that the game took place in a city, that it was a T-rated game—but beyond this, we weren't sure of much. We described the game as a combination of ANIMAL CROSSING, THE SIMS, and GRAND THEFT AUTO—an action game with pervasive simulation elements. It isn't surprising that this description didn't exactly enlighten people—we should have realized that our inability to articulate a crisp vision for the game meant that we didn't have a clue what we were doing.

More than anything, this lack of focus represented us misapplying the prototype-test-iterate cycle that served us so well in tightening up gameplay. We worked on prototyping a bunch of innovative little gameplay scenarios, expecting that a clear vision of what the game should be would arise from the prototypes. It didn't; we just ended up with a bunch of little prototypes that had nothing to do with each other. We were nearly a year into the development of what was ostensibly a combat-focused action game, but still hadn't done any work on letting the player fire a weapon, or on having enemies that shot back!

Eventually, we wised up, realized how lost we were, decided that we were doing an action game, and started working on basic game play. Things progressed quickly from there; most of the innovative little gameplay scenarios were set aside, to be revisited once the core of the game was tight. A few of them even ended up in the game—like pedestrians spontaneously taking pictures of the hero. Overall, though, our lack of focus cost us months and months of work, pursuing what ultimately turned out to be discarded ideas.

GAME DATA

inFAMOUS

PUBLISHER SCEA

DEVELOPER Sucker Punch

NUMBER OF DEVELOPERS 68 at peak; 5 external contractors

LENGTH OF DEVELOPMENT 3.5 years

RELEASE DATE May 2009

TOTAL LINES OF CODE 850K game; 125 K script; 350K tools + pipeline

PLATFORM PlayStation 3

2 } Underestimating the Cost of Realism. We wildly underestimated how difficult the transition would be from the bright and cheerful PlayStation 2 graphics of the SLY COOPER games to the gritty and realistic PlayStation 3 look of INFAMOUS. In effect, we tried to take two steps at once. First, we needed to catch up with the realism in the best-looking PlayStation 2 games; second, we needed to match the even higher quality bar being set by the best teams as they moved to the next generation of hardware.

This was a struggle on multiple levels. First, we had a technology gap to clear; the engine we used for the SLY COOPER games needed to be substantially rewritten to run on the PlayStation 3. We needed to rework our whole pipeline to accommodate things like vertex and pixel shaders. Unfortunately, we didn't start this effort until after we completed SLY 3, leaving us without a pipeline or engine running on the target hardware for the first ten months of product development.

The lack of a working pipeline and engine made it a challenge for our art team to learn the new skills and techniques necessary to build the realistic content we needed for INFAMOUS. All our assumptions about how to build content needed to be revisited to match our new expectations and the capabilities of the new hardware. In addition, our art team needed to hone their skills building realistic content, and the lack of a working engine made this painful.

3 } Not Enough Polish Time. Despite our best efforts, we ended up with less polish time than we needed for INFAMOUS, and the game suffered as a result. If we'd had another month or two of dedicated polish time at the end of the game, the whole experience would have been much smoother. It's

a mistake we've committed on every game we've made, so there are clearly cultural issues at work.

It's pretty hard to cut the amount of content in the game, but in retrospect that was what we needed to do. Until we were very close to shipping, we weren't sure how long a single play-through of the game would take. We had some data from focus groups and from our test teams, but we didn't trust it. In addition, we weren't sure how much of an effect optional content like mini- missions would have, and we weren't sure how many players would choose to replay the game, either to experience the other available karmic path, or to try the game at a different difficulty level.

We feared the game being too short, and were afraid to cut missions and story content from the plan. In addition, we generally implemented

content and missions in the order in which the player would encounter them in-game. Toward the end of development, we were boxed in, needing to implement a few more missions to wrap up the game's storyline.

In the end, this was mostly a failure of planning. We also suffered from not driving any slice of the game to a truly "shippable" level of quality early on. If we'd done this, we would have had a much better idea of how much polish work remained to be done on the rest of the game.

4 } Melee. From the start, we thought of INFAMOUS as primarily a third-person shooter. We took a while to figure out exactly how the player's attacks would work—at various times, the main character Cole carried a pistol, or threw sharpened bike gears, or telekinetically grabbed nearby stop signs and hub caps and chucked them at enemies, before we finally settled on the electrically-themed attacks in the shipping game—but we always believed that aiming and shooting were the core of the combat experience.

As a result, we didn't pay a lot of attention to hand-to-hand combat. We supported a simple "punch and kick" attack system, but didn't work on it very hard. It didn't look good, and it wasn't fun to use, but we figured it wasn't important, since players wouldn't get close enough to enemies to use it.

This attitude lasted until our first external focus test. Many players approached the game as we intended; they looked for cover, popped off long-range shots at bad guys, and waited until enemies were down before proceeding.

Lots of players, though, dive-bombed at a full run into every group of enemies they found, expecting to use their melee attacks to express

postmortem postmortem

their karmic choices. These players had no fun at all—they were peppered with shots when trying to rush groups of enemies, then were frustrated when their melee attacks misfired due to our under-engineered melee system. Worse, the strategy worked just well enough that punch-happy players stuck with it, even though they weren't having any fun.

As professional game developers, we knew exactly how to fix the problem. The problem, we decided, was that we weren't trying hard enough to dissuade players from using melee. We fixed this in the game's AI: we made enemies more accurate at close range, we had them back away from nearby players, and so on. Problem solved!

Until the second focus test. Unfortunately, many players still didn't take the hint; they tried to punch and kick their way through every combat. Even with the revamped AI, they could get close enough to throw a punch, but the melee system was bad enough that they couldn't count on it landing. Result: unhappiness.

Our fundamental mistake was to assume players would understand how we expected them to play the game (and then want to do it); our secondary mistake was to solve the mismatch by trying to force them to play the game the "right" way. It turns out that many people assume that a game featuring a superhero will involve super-heroic punching and kicking, and that in fact super-heroic punching and kicking is what they like about superheroes.

By the time we figured this out, it was too late. We had time to improve the melee system so that it looked good and played smoothly, but we didn't have enough time to make major changes to its role in INFAMOUS. We ended up with a melee system that was fun, but only useful in limited circumstances. Kicking a bad guy off the side of a building is a blast, but it doesn't happen often enough during normal gameplay.

5 } Ratings and Cultural Sensitivities. INFAMOUS was our first foray into T-rated content, and we were unsure of our footing. The game is pretty open-ended, and it wasn't clear exactly what we could let the player do, especially since the game world contained innocent civilians as well as hardened enemies. An attack which felt justified against a gun-toting thug felt disproportionate when deployed against a woman running away from you. Giving the player an opportunity to make ethical choices was a crucial component of the game, but keeping the game within T-rated bounds was important to us.

Even our attempts to provide the player with non-lethal weapons caused problems. In INFAMOUS, it's possible to heal a wounded character. Initially, this seemed like a pretty safe and wholesome ability, until focus test players started farming experience by shooting civilians, healing them,

then repeating. Similarly, a fair-minded player can "hog tie" other characters to the ground, restraining them without killing them. Again, this seemed reasonable, until players started zapping the hog tied character, just to see them writhe in agony.

Our attempts to give people real choices about how to play the game made it possible to be pretty nasty. Reconciling this with our goal of a T rating was a trick, with delicate compromises necessary in many parts of the game. Nevertheless, we ended up with a game that was rated T ... in the U.S.

When we started submitting the game for rating outside the U.S., our delicate fabric of compromises fell apart. Different cultures have different sensitivities, and what seemed reasonable to a U.S. audience seemed less so overseas. For instance, in our original builds it was possible to kill civilians; due to European ratings board concerns, we made "death" a strangely non-permanent state for civilians. They'll drop motionless to the ground when shot, but after five minutes they'll recover, stand up, and walk off.

In other cases, the problems were unsolvable. In Japan, the ability to harm pedestrians was in itself enough to earn the game a Z rating, restricting its sale to 18 year olds and up.

A FICKLE FOOD UPON A SHIFTING PLATE

>> Like a lot of game projects, INFAMOUS went through some serious twists and turns in the course of its development. Our original vision for the game—ANIMAL CROSSING played as a superhero—is nearly unrecognizable in the final game. And yet, we're happy with the game we shipped, and proud of what we as a team accomplished.

None of this would have been possible without the support and encouragement of our publishing partners at SCEA, and without the dedication and hard work of everyone at Sucker Punch. For INFAMOUS to succeed, everyone at Sucker Punch had to step up, learn new skills, and embrace the challenge of doing something entirely new. And they did.

We certainly didn't take the straightest path to success, though; it seems like whenever we think we've got this whole game development thing figured out, we end up discovering how little we really know.

But I'm sure everything will go smoothly on our next game. 🔴

CHRIS ZIMMERMAN *helped start Sucker Punch in 1997, after working for a long time at a software company too large and powerful to risk mentioning. He's been the development director on all of the Sucker Punch games. It's been rumored that he was also the guy who suited up for all of Cole's motion capture sessions, but this has no basis in fact. Email him at czimmerman@gdmag.com.*

AUTODESK
3DS MAX 2010

REVIEW BY JEFF HANNA

AFTER THE SHORTENED DEVELOP- ment cycle, earlier than normal release, and the limited number of new features in 3ds Max 2009, many people were questioning whether Autodesk was still willing to offer feature-rich upgrades to the package after purchasing it. The recent release of 3ds Max 2010 answers that question with a wholehearted, "Yes!" There are compelling new features and welcome enhancements in just about every major area of the software.

USER INTERFACE
>> At first glance 3ds Max 2010 looks a bit different from earlier releases. For one thing the new default color scheme is dark, like many other Autodesk products. Another change to the UI is the addition of a Microsoft Office 2007-esque Ribbon, located under the menu and toolbar. The Ribbon

updates dynamically to show tools pertinent to what the user is doing at that moment, which helps expose new features when they are needed.

Adding the new Graphite modeling tools (discussed later) into my workflow was easy because the Ribbon never overwhelmed me with options. Unfortunately, some aspects of the Ribbon are unnecessarily awkward. For instance, the button to open the material ID assignment dialog box is located on the right end of the Ribbon, but the dialog box always opens at the left end of the Ribbon, requiring that the mouse be moved to the opposite side of the 3ds Max window. Autodesk could've reduced extraneous mouse movements by having the Ribbon open floating dialogs on, or near, the Ribbon tools that call them. Even though I like the Ribbon I find

the Max UI is becoming cluttered, with a menu bar, tool bar, and now the Ribbon. Hopefully in future releases Autodesk will find a way to unify these.

MODELING TOOLS
>> The popular third party polygonal modeling tool PolyBoost has been integrated directly in to 3ds Max 2010 and is now referred to as Graphite. Graphite has tools for freeform polygonal sculpting with a wide variety of brushes, quickly re-topologizing a mesh, locking transforms to any surface on an object, creating and modifying smart selections, and drawing surfaces and shapes. The 100-or-so new modeling tools Graphite adds to 3ds Max expands an already impressive polygonal modeling toolbox with excellent new features.

3ds Max 2010 also introduces xView geometry checking. XView

is a rather comprehensive system for checking geometry for errors. XView checks for a staggering number of geometry errors such as; T vertices, unwelded vertices, and flipped faces among others. XView is exposed to both the SDK and MaxScript so that it can be factored directly into custom tools. Having a powerful geometry checker as part of a standard modeling workflow will reduce downstream geometry problems in any pipeline.

ProOptimizer is another welcome new feature in 3ds Max 2010. It takes over for the aging MultiRes modifier. ProOptimizer is a powerful polygonal reduction tool. Artists can control every aspect of the polygonal reduction, from the number of final faces to which faces should be removed last, and even if UVW layouts should be respected. Bringing high resolution Mudbox or ZBrush files into 3ds Max to create normal maps and reduce the geometry down to game resolution has never been easier.

SCENE MANAGEMENT
>> Even though the new Containers feature enables sharing of data from one 3ds Max scene file within a different scene file, it's wrong to call Containers a new version of XRefs. Yet, there are parallels. Containers offer many new ways to manage the data within a scene and easily collaborate with others. Multiple objects can be gathered up and placed in a Container. Objects in a Container can be treated as one entity. If scene performance is less than optimal, Containers can be unloaded from the viewport, while maintaining connections the objects in the Container have with the rest of the scene. Containers that are unloaded from the viewport will

3ds Max 2010's Graphite tool enables freeform polygonal sculpting.

not be immediately loaded when that scene is opened in 3ds Max 2010. This can reduce load times on complex scenes and allow the user to selectively load only the portions of the scene that they need.

Like groups, Containers can be copied, repositioned within the scene, and deleted. Any objects within the Container will inherit whatever actions are performed on the Container. Containers can be referenced in other Max scenes. Containers maintain permission lists that manage the access to, and editing of, the data within a Container by other users. For me, the Containers toolset in 3ds Max 2010 is one of the most compelling new features. XRefs were a promising, but frustrating, way to share data. Containers address the shortcoming of XRefs and also offer new ways to share data and manage complex scenes.

MATERIALS AND MAPPING

» In previous versions, 3ds Max could only display DirectX shaders in a viewport, not render them. Converting DirectX shaders for use in a rendered scene has always been the job of "that" guy at work, the one who came back from GDC five years ago with a book on mental ray that he's never really read. Whenever marketing calls and requests high resolution renders of game assets he's the one who has to re-arrange his schedule and frantically translate shaders. Autodesk has solved this problem (or possibly ruined this poor soul's hobby) in 3ds Max 2010 by including mental mill. Mental mill is a schematic shader editor that creates shaders in the MetaSL language. MetaSL shaders are hardware agnostic and mental mill can save shaders as HLSL, CGFX, GLSL, as well as the mental ray format. Even if you prefer authoring DirectX shaders in another application, mental mill's wide array of shader features will make it easy to create equivalent shaders for use in renders.

The Material Explorer is another new and welcome feature. Like the Scene Explorer before it, the Material

Explorer displays all materials used in a scene in a filterable table view. Listing all of the properties of numerous multi-sub materials simultaneously gives the artist a complete view of, and great control over, the materials being used in the scene. For viewing and managing large numbers of materials in an easy to understand format the Material Explorer is superior to the Material Editor in every way.

The new Viewport Canvas feature provides the ability to

paint in a viewport and have that information mapped to a selected object. When Viewport Canvas is active the currently active viewport becomes a 2D texture canvas. After the user has finished painting, the canvas is stored as a texture and slotted into a material. While Viewport Canvas is not a dedicated 3D painting program and is not designed to be one, it is useful for quickly blocking out UVW space and base colors on an object.

RENDERING

» The 3ds Max viewports have been steadily gaining real-time rendering abilities over the past few releases. Autodesk refers to these new features collectively as Review. For 2010, Autodesk introduces HDRI lighting, soft shadows, ambient occlusion, hardware anti-aliasing, and interactive exposure controls. These new capabilities, along with MetaSL shader display, give the 3ds Max viewports the ability to present an image or animation preview that is a very accurate approximation of how the final rendered data will appear.

The new gamma correct workflow will ensure that those renders will display with accurate color. Gamma settings are read and maintained for each loaded texture and are properly displayed in the viewports and communicated to network rendering machines. For industries where output color accuracy is critical this new workflow is very welcome.

SPECIAL EFFECTS

» 3ds Max 2010 sees one of the first major enhancements made to ParticleFlow since it was released. PFlowAdvanced includes fourteen new operators including more precise particle placement tools, grouping operators, and new ways of defining particle shapes. The UI complexity of PFlow has been reduced and the performance has been increased. 3ds Max 2010 ships a new library of over one hundred high quality pre-built particle effects. For those instances when a render or animation needs some

effects to spruce it up and time is tight, the new effects library will be very useful.

MAXSCRIPT AND THE SDK

» MaxScript has been enhanced with two language improvements and a new debugger feature. MaxScript now benefits from having the ability to label struct members as either `public` or `private`. This furthers the ability to use structs as almost class-like constructs. MaxScript now also includes an `assert()` method. This makes it trivial to validate input or generated values and alert the user if something is amiss. Both `public`/`private` and `assert()` will help make MaxScript based tools more stable and less error prone.

The MaxScript debugger has been beefed up and can now show the line number where a problem was encountered in a script. This new feature is welcome, but does not go far enough to elevate the functionality of the debugger to the point where it would truly be useful. Until Autodesk includes visual breakpoints in the MaxScript editor and allows line by line code stepping, including stepping into or over functions, the debugger will remain a promising but underpowered tool.

CONCLUSION

» The wide ranging new features are all worthwhile enhancements to the package. There are new tools that will improve the workflow of just about everyone involved in modeling and animation. 3ds Max 2010 offers excellent stability and increased performance compared to previous releases. For me this is the most compelling 3ds Max release in quite some time. Any current user, or anyone looking for a feature packed modeling and animation package, should at least take a look at 3ds Max 2010.

JEFF HANNA is a senior technical artist at THQ's Volition studio, where he recently shipped RED FACTION: GUERRILLA, and is a member of the Game Developers Conference Advisory Board. You can reach him at jhanna@gdmag.com.

BECOMING SOCIAL

LEVERAGING SOCIAL NETWORKS IN YOUR GAME

ONLINE PLAY WAS ONCE A RISKY bonus. As broadband penetration has increased though, it has come to be an expected feature in many kinds of games, usually in the form of multiplayer. There's a lot more that can be done online beyond connecting players based on their skill level, spawning them in a level full of weapons, and letting them go at it killing each other.

The last couple of years have seen the bloom of social networks, Facebook and Twitter in particular. Both platforms have open APIs that allow us to integrate them in our games. Web games were the first to embrace these social networks, followed by mobile games. PC games are starting to catch on, but unfortunately console games haven't jumped on the bandwagon because they have their own proprietary friend/social networks and usually have strict restrictions about what content can be accessed online. Consoles were also late to the online gaming party, so we can only hope they will continue evolving and will soon embrace social networking in both their portals and inside games.

WHY SOCIAL NETWORKS?

>> You may think that if you're not making a Facebook game, why do you need to care about social networks? Actually, unless you're making console games on closed network platforms, the question becomes: How can you afford not to be integrating social networks in your games?

One of the most basic forms of sharing things about a game with other people is the tried-and-true online leaderboard. That idea has been around since the days of stand-up arcade games, so it's nothing new. But most players will not be one of the top-10 players in the world at any one game, so a global leaderboard is pretty meaningless to the majority of people. And somehow, knowing that you've moved from position 37,039 to 37,038 just doesn't sound very exciting. That's why we have friends-only leaderboards.

But we don't need to stop there. What games are doing today is just the beginning of the possibilities that social networks bring to games. We could fetch data from our friends and integrate it in the game. Maybe bring in their names or pictures and use them on in-game characters. Or maybe the game even contains some elements that require you to interact with your circle of friends directly. We'll be discovering more effective ways of social integration in games in the next few years.

SOCIAL APIS

>> No matter which social network you end up working with, they all have a lot of things in common. They all want you to integrate with their network because your game will make it even more attractive for more people to join, so they all provide free APIs you can start using right away. No waiting, no approval. Sometimes you need to sign up with their developer program, and sometimes you can start using them immediately.

Being online social networks, their APIs are all based on HTTP calls, and usually follow the REST architecture. To call a particular API method, you need to make an HTTP request to a particular server, and pass any arguments as GET or POST parameters depending on the API. The responses

listing 1a example response in XML

```
<?xml version="1.0" encoding="UTF-8"?>
<friends_areFriends_response
    xmlns="http://api.facebook.com/1.0/"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="..."
    list="true">
    <friend_info>
        <uid1>222332</uid1>
        <uid2>222333</uid2>
        <are_friends>1</are_friends>
    </friend_info>
    <friend_info>
        <uid1>1240077</uid1>
        <uid2>1240079</uid2>
        <are_friends>0</are_friends>
    </friend_info>
</friends_areFriends_response>
```

listing 1b example response in JSON

```
[
{"uid1":"222332","uid2":"222333","are_friends":true},
{"uid1":"1240077","uid2":"1240079","are_friends":false}
]
```

from making the API call are usually in either XML (Listing 1a) or JSON (Listing 1b) formats. In either case, they're both human-readable text formats, which makes debugging them fairly straightforward.

Because method calling is all done through HTTP requests, everything is asynchronous. This means it's possible for the server to be down, or the internet connection to be interrupted, so it's important that the code you write handles those situations correctly.

If the thought of hand-crafting HTTP requests and parsing responses doesn't seem like a lot of fun, most APIs have wrappers in a variety of languages. Check the developer site and the user contributions and you'll probably find a library that wraps all those `HTTP` requests into something more natural in C++ or whichever language you're using in your project.

FACEBOOK
>> There's no doubt that Facebook is the current heavyweight champion of the social networks, with over 120 million unique visitors per month. This is a case where success feeds on itself, and because so many people are already on Facebook, it's more likely for their friends to join. The social graph of friend connections you can retrieve from Facebook is going to be quite extensive.

The Facebook Connect API matches the expansiveness of Facebook itself, with lots of different functions offering lots of functionality, from gathering information about your friends, to uploading pictures, or posting links on someone's wall. At this moment, the API contains 104 different methods.

For example, to set the user's status, you need to call the function `Users.setStatus` with the following required parameters: `api_key`, `call_id`, `sig`, `version`, and the following optional parameters: `session_key`, `format`, `callback`, `status`, `clear`, `status_includes_verb`, `uid`.

Unfortunately, as soon as you look underneath the hood, Facebook stops being so attractive. The API shows clear signs that Facebook went through an accelerated growth period and that architectural concepts were very much in flux. There are multiple API calls to do almost the same thing, with slightly different side effects. New calls are being added on a regular basis, and existing ones are tweaked and their functionality changed. As game developers, we might be used to that in our own code, but it takes some adjusting to dealing with that kind of design from an external API we need to rely on.

You also don't have full control over how the results of the actions performed through the API will appear to other users. Things will look radically different depending on whether you're browsing Facebook through the web, a mobile device, or the iPhone.

To get started with Facebook development you need to sign up as a registered developer (no approval process, just sign up and go). Perhaps one of the most confusing aspects is that even if you're using Facebook Connect in your own game, you need to create a dummy Facebook application that does nothing. Part of the baggage of the organic architecture it seems.

To allow a user to log into Facebook from within your game, you need to use their custom login web form directly. From a security point of view, that's a pretty logical approach because it prevents the game from possibly storing the Facebook username and password of the user (which might be hacked more easily than Facebook itself). On the other hand, it means that the Facebook login screen is not well integrated with the game and it can be quite a jarring change from the rest of the game visuals.

TWITTER
>> If Facebook is the heavyweight champion, Twitter is the up-and-

coming contender. By last count it has "only" around 23 million unique users per month, but it's growing at a faster rate than Facebook. Twitter is also designed to be much simpler and lighter weight than Facebook. Your range of actions are pretty much limited to writing 140 character messages, updating your friend network, and performing searches.

When considering integrating Twitter in a game, it's important to realize that it's also quite different than Facebook from a design point of view. Twitter is more about real-time exchange of information, and things are a lot less permanent than in Facebook. It might make sense to post a picture of an item you just unlocked, but there's very little point in creating an album of pictures showing some in-game story because nobody will see it

in a few weeks. Things older than a couple of days are ancient by Twitter standards.

The simplicity of Twitter is reflected in its API as well. It's much smaller and cleaner than Facebook Connect (47 different API calls), but it's also more limited in scope. As a result, the API is also much more stable than Facebook, and it's much easier to integrate in games.

For example, to set the user status, you call the function `statuses/update` through `http://twitter.com/statuses/update.format` with only two parameters: `status`, and `in_reply_to_status_id`.

URL SHORTENERS
>> Naturally, people like to share links to different things around the web. Your game might also

provide links to high-score tables, user game profile pages, unlocked achievements, and so forth. Long links can be cumbersome to work with and a sore on the eyes. Additionally, since Twitter is limited to messages that are 140 characters in length total, including any URLs in the message, it's no surprise that URL shorteners are a crucial part of the Twitter (and to a lesser extent, Facebook) culture.

A URL shortener provides a short URL that is an alias for any arbitrarily long URL. The short URL resolves into a server, which then redirects the request to the final URL. Usually, these URLs are permanent and will not change (as long as the company providing the URL shortening service is still around).

Some of the most popular services are tinyURL, bit.ly, and post.ly. They all offer an open API to create new URLs, that is so simple that I hesitate to even call it an API. For example, to create a short URL with tinyURL all you need to call is `http://tinyurl.com/api-create.php?url=[your_url_here]` and it will return the shortened URL in plain text. Can't get any easier than that!

PICTURE SERVICES
>> Because of the simplicity of Twitter, a variety of third-party services have sprung up trying to let people use Twitter and other social media in new ways. One of the most popular services allows users to quickly upload a picture from anywhere, which can then be shared on Twitter.

The most common picture service is TwitPic, and, like the URL

shorteners, it's extremely focused on doing just one thing. It only has two API methods: `http://twitpic.com/api/upload` and `http://twitpic.com/api/uploadAndPost`, and the latter is just a convenience method combining the upload and putting a Twitter message status. TwitPic takes the easier route by using the same username and password as Twitter, so there's no extra account information to type.

Another picture service, yFrog, aims to have a bit more functionality, and provides support for generating thumbnails for example.

HOW NOT TO INTEGRATE SOCIAL NETWORKS
>> A few months ago, armed with this knowledge, I decided to add Facebook integration to my iPhone game, FLOWER GARDEN. The Facebook Connect API has the steepest learning curve of all the social networks I've looked into, but after a bit of work, I was able to log in, retrieve all the friends, their picture, upload an image of the bouquet, and post it to a friends "wall" directly.

There was a lot of getting up to speed with different concepts (feeds vs. walls vs. stories), creating development test accounts, and exploring the API to find the most appropriate way to implement that functionality. I even had to get my app whitelisted by Facebook to be allowed to write to a friend's wall. But I eventually got it working in a bit over a week. Naively, I thought I was done.

What I failed to realize is that the Facebook API and the implementation itself is a state of flux. Things that are working today, stop working tomorrow, and then work again a day later. My first reality check came a couple of days after releasing the FLOWER GARDEN update with Facebook integration. I found out that new Facebook users couldn't log in and send bouquets. Puzzled, I spent a while debugging the problem until I discovered that Facebook had changed the format of their user IDs, breaking their own

» Zynga's YOVILLE is playable on Facebook and MySpace.

library wrapper and every iPhone app with Facebook integration out there in the process. From the moment I fixed it, to the moment the update was available, two weeks had gone by with users upset they weren't able to use Facebook.

A few weeks later, out of nowhere, posting a bouquet to someone's wall stopped working with an error from Facebook. As far as I could see, nothing had changed, and some other developers were also reporting problems in their games. We submitted it as a high-priority bug to Facebook, and they fixed it 48 hours later. The reason: Apparently some Facebook developer had checked in some code changes without running unit tests!

Maybe this constant change is not a huge deal if you're writing a web application. As soon as a problem arises, you can throw resources at the problem and come up with some fix on the server relatively quickly. It's still a drain of money and time, but you can deal with it. As a PC or mobile developer, the situation is much more serious because you can't push an update instantly to all your users. And as a game console developer, the problem

might simply be fatal since you might not be able to get an update for months because of the QA and approval process involved.

Does it mean we should give up on social network integration? Not at all. The solution is to avoid integrating it directly in the game like I did, and instead add a level of indirection. Have your game communicate directly with your server, and your server can call the social network API using PHP (or whatever your favorite language is). If anything changes, you can quickly update your server to keep everything working. In the worst situation, you can always temporarily shut down the social network integration and notify your user in some pleasant way instead of getting cryptic errors directly from the social network API.

Social networks are here to stay. I'm really looking forward to seeing how things will evolve and what new ways the industry will come up to integrate social networks in games. 🎮

NOEL LLOPIS has been making games for just about every major platform in the last ten years. He's now developing games for the iPhone from local coffee shops. Email him at nllopis@gdmag.com or follow him on Twitter @snappytouch.

SEND IN THE CLONES

GETTING THE MOST OUT OF PAPER DOLL SYSTEMS

CHARACTERS ARE THE MOST expensive and finicky assets in any game, both in terms of production costs and runtime budgets. Being so expensive, of course, there are never enough of them to go around. As a result, we've become sadly resigned to the notion that entire economies are serviced by sets of twins, whole cities are inhabited by just the cast of *Jon And Kate Plus Eight*, and mighty armies are staffed entirely by clones (and just for the record, George, we have an excuse. You don't!)

Nobody likes killing the same minions, chatting to the same civilians , or ordering the same flunkies to their demise ad nauseam. The standard tactic for disguising the clone-ification of our worlds is to build characters with a library of swappable pieces and color-shifted textures. By mixing and matching these pieces we hope to create visual variety with a minimal expenditure of time and runtime resources.

Unfortunately, designing and managing these paper-doll systems is a daunting task. Art-directing a bunch of disparate elements and hoping they'll come together into an appealing, coherent whole is hard. Creating individual elements that combine well visually and can still animate without clipping is hard. Creating tools to let artists efficiently create and try out combinations of bits so they harmonize is hard. Hell, just managing all the pieces efficiently is hard ("remind me again which one '`right_shoulder_fancy_leather_brassard.ma`' is?") To top it all off, serious paper doll sets can be very large and complex. RPGs, in particular, need literally hundreds of variants of clothing, armor, and decorations for each distinct race. Player demand for new outfits and new equipment never lets up.

Since every game with a paper doll system wrestles with knotty budgetary and scheduling problems, it's important to have some idea of which strategies work best—which assets do the most to promote the illusion that your little cast of clones is really a rollicking crowd of individuals? It's infuriating to reach the end of a year's worth of building capes and shoulder pads only to discover your new player race really looks like a squad of clones with highly diverse shoulder pads. Luckily for us, though, this problem has started to attract some serious scientific interest from researchers who study human perception. Their work provides some useful guidance for teams that want to design paper doll systems that really stretch their limited resources effectively.

METRO
>> The *Metropolis Project* at Trinity College in Dublin, Ireland, represents the work of a research team studying strategies for large-scale virtual urban environments with realistic virtual crowds. This kind of project involves many of the same problems we encounter in games—particularly when it comes to creating large, visually diverse crowds within the constraints of authoring time and runtime budgets. But the most valuable contribution of the project isn't some kind of new compression algorithm or LOD strategy. Instead, the team combines game-like programming tactics with psychological research into what creates the feeling of uniqueness and variety we're all after.

So, in a series of studies over the last few years, researchers from the *Metropolis Project* have been trying to quantify the effectiveness of many of the hacky techniques we in the games business have always used to create variety in crowd scenes. Building effective paper doll assets is, as we've said already, a tough job; having some objective guidance in this tricky venture is a big step forward. The original research can be interesting to read (see sidebar for references) but to simplify things a bit for those of us who slept through statistics it makes sense to summarize the results.

FACE EFFECT
>> The most notable results for our purposes came from a study in which researchers used eye-tracking devices to follow the gaze
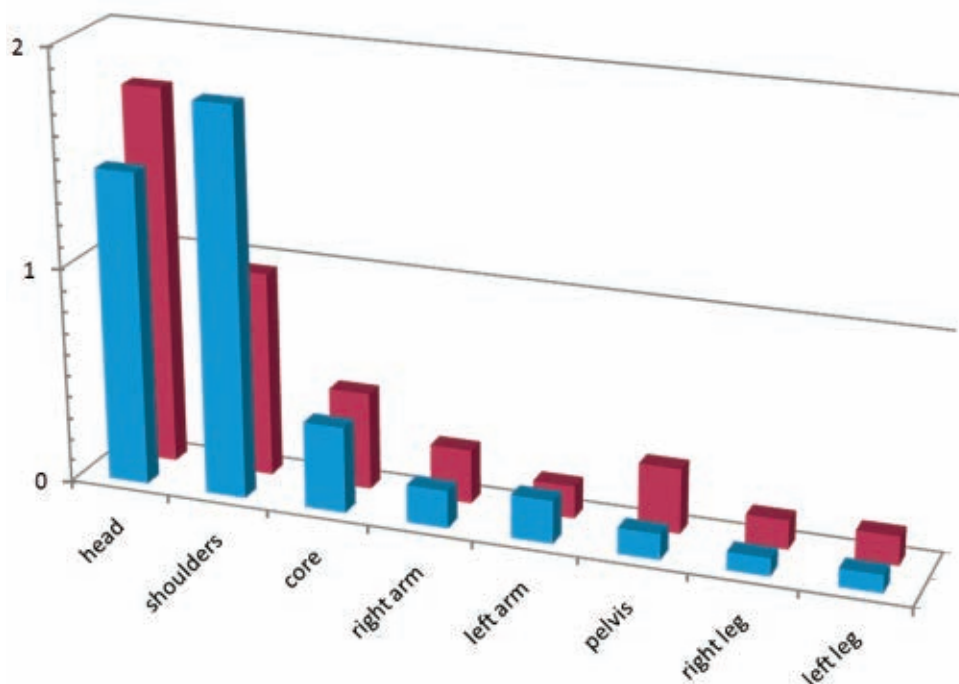
FIGURE 1 This chart shows the average length that viewers in the *Metropolis Project* study focused on different body parts when trying to pick clones from a static (blue) or moving (red) crowd of models. Times in seconds (Adapted from McDonnell, Larkin, et al, *Eye-catching Crowds: Saliency Based Selective Variation*. ACM Trans. Graph. 28, 3 (Jul. 2009)).

FIGURE 2 An example of how faces are the most important element in differentiating a crowd. The company of soldiers, though dressed identically, stands out as a group of individuals, while it's easy to see that mannequins in a store window are really identical, despite their varied outfits.

of participants trying to identify cloned characters in a line-up. This is a common tactic in the science of human perception—seeing where the eye lingers gives a very good idea of what the brain thinks is important about an image. In this case, when the participants in the study were asked to pick out identical clones mixed in with unique models, researchers found that they always concentrated very heavily on the face and upper body. As you can see in Figure 1, the head alone got as much attention from viewers as the arms, legs, hands, and feet combined.

This isn't really surprising, when you think about it. Any artist knows people are extremely good at recognizing faces. It's precisely that ability to hone in on subtle details of expression and appearance that makes it so difficult for our characters to avoid the "uncanny valley" effect (see December 2004 Pixel Pusher).

So, we know that faces are the first, most important thing viewers look at when seeing a person for the first time. What's interesting, though, is that it seems the importance of faces creates a kind of "magnetic attraction" which repeatedly pulls the eyes back to the head, shoulders and upper torso, while the rest of the body doesn't get much attention.

HEAD O' THE CLASS

>> This focus on faces is a supremely important point to remember when designing a paper doll system. Heads and faces are, by far, the most distinctive elements in any crowd. This makes a lot of sense, if you think about the images in Figure 2: Line up a squad of real people in identical clothes and they remain a collection of distinct individuals, but a fill a store window with mannequins in diverse outfits and it's easy to see they're just clones. Perceptually, at least, the character is the face. If you're really interested in making your world seem more populous, add more heads!

Unfortunately we all know this is a pricey proposition. Even in the era of Z-brush and Mudbox, good faces are hard to do well. (For some suggestions how to streamline your head pipeline, see "Face to Face" in the March 2008 issue). It is, however, possible to stretch limited resources procedurally. Games like FALLOUT 3 and OBLIVION use fully parametric faces, which provide an enormous range of appearances for fairly low runtime costs. A less ambitious tactic is to adopt common head topology and UV layout and then blend face textures at runtime. This enhances visual variety without completely busting the memory bank. Blending even works—within limits—on normal maps.

It's true that neither parametrically generated nor blended faces can really compete with carefully hand-crafted ones. Artists love to roll their eyes at some of the results of these

techniques (you can certainly find some astonishing howlers from Bethesda fans on YouTube!). Still, in the context of the Dublin study it seems almost anything you do to increase the range of faces your players see, the better—even if it may involve trading off some loving craftsmanship for a broader scope.

HATS!

>> But even if you limit yourself to simply adding more heads to your existing lineup, the facial fixation effect has other implications you should think about. Since the eye lingers in the vicinity of the all-important face, anything close by assumes greater importance for the viewer.

When researchers compared the effectiveness of different kinds of part swaps for disguising clones, they found that accessories and clothing pieces were much more important if they were near the face: hairstyles, hats, and glasses turn out to be very important for creating the illusion of diversity. Individually each of these items has

a similar effect on viewers' ability to spot clones. Interestingly, though, the combination of hairstyle and glasses together was noticeably more effective at disguising clones.

CLOTHES

>> On the flip side, that relentless tendency of fixating on the head and upper body means that some kinds of variants don't help much in differentiating characters. For example, different pants or shoe styles won't have much effect on your players' perceptions of variety (a fact which, incidentally, provides scientific closure to a decade's worth of arguments between male and female viewers of *Sex and The City*. It turns out your $900 Jimmy Choos really don't help you stand out in a crowd. Sorry.)

This doesn't mean that your paper doll needs to consist of a single jumpsuit with a ton of hats, beards, and glasses. Clothing plays an interesting and ambiguous role in paper doll systems for games. The Dublin study suggests that clothing plays an important, but

not a dominant role in clone-spotting. Eye-tracking data did show, fairly conclusively, that shirts and jackets received much more attention from viewers than pants, skirts, or footwear. (see Figure 3). As a method of disguise, changing outfits does work to some degree. However, as any paparazzi can tell you, people can still spot a familiar face whether you're in a t-shirt or a tuxedo—that's why celebrities in search of privacy invariably adopt the hat-and-sunglasses strategy recommended by the researchers. (And who'd have guessed that Lindsay Lohan spent her spare time reading SIGGRAPH papers?)

Of course we usually need each a character's outfit to convey important gameplay information, and that's outside the scope of academic work. Your players want to know by sight if a character is friendly or an enemy, lightly or heavily armored, healthy or damaged, and so on. All these issues fall outside the scope of the academic research and into the laps of game designers. The Trinity College

studies were deliberately focused on mundane, everyday crowds wearing appropriately casual clothing.

Color changes, like clothing swaps, are strongly tied to gameplay needs and help create visual variety. Color shifts are fast and cheap and, like clothing types, they help slightly to disguise otherwise identical models. As you can see from Figure 3, however, they are only mildly helpful. Nevertheless it's always a good idea to include color shifting in your paper doll toolkit. It can enhance gameplay (by strongly identifying classes of characters, a la HALO) or simply add more visual variety to your world for very little cost in memory, which makes it a good bargain even if it can't alleviate clone monotony.

NO SILVER BULLET

>> Sadly, the research from the *Metropolis Project* isn't a magic formula for creating effective paper dolls. But it is very useful ammunition for the artist who has to quickly fill the streets of a virtual world on a limited budget of time and RAM. It's also a good example of why it's sometimes handy to geek out at conferences like SIGGRAPH or GDC—those charts and formulas might seem light-years away from artistic concerns, but academic research like this can often be a serious boost to even the craft-oriented artists. A good bit of research makes us articulate and clarify the techniques we sometimes evolve by artistic intuition. In this case, the Dublin research helps us understand some of the reasons we react to the art in games we like and also helps us understand why some things (those Jimmy Choos!) never added up. If that means putting up with a few graphs, it's a price worth paying. [GD]

STEVE THEODORE *has been pushing pixels for more than a dozen years. His credits include* MECH COMMANDER, HALF-LIFE, TEAM FORTRESS, COUNTER-STRIKE, *and* HALO 3. *He's been a modeler, animator, and technical artist, as well as a frequent speaker at industry conferences. He's currently a consultant helping game studios perfect their art tools and pipelines. Email him at* stheodore@gdmag.com.
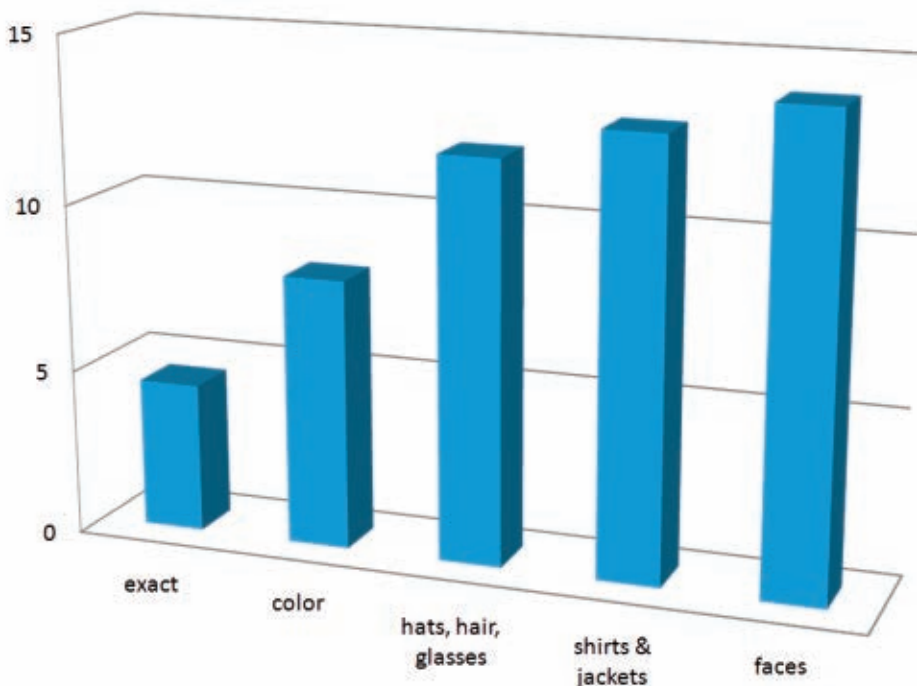
>> FIGURE 3 The effect of different kinds of variations on the time it takes to spot a clone (times in seconds). Color variations and accessories are some help in disguising clones; texture changes on the torso and above all faces are the really important factors.

Talented Programmers Apply!

WWW.NEVERSOFT.COM
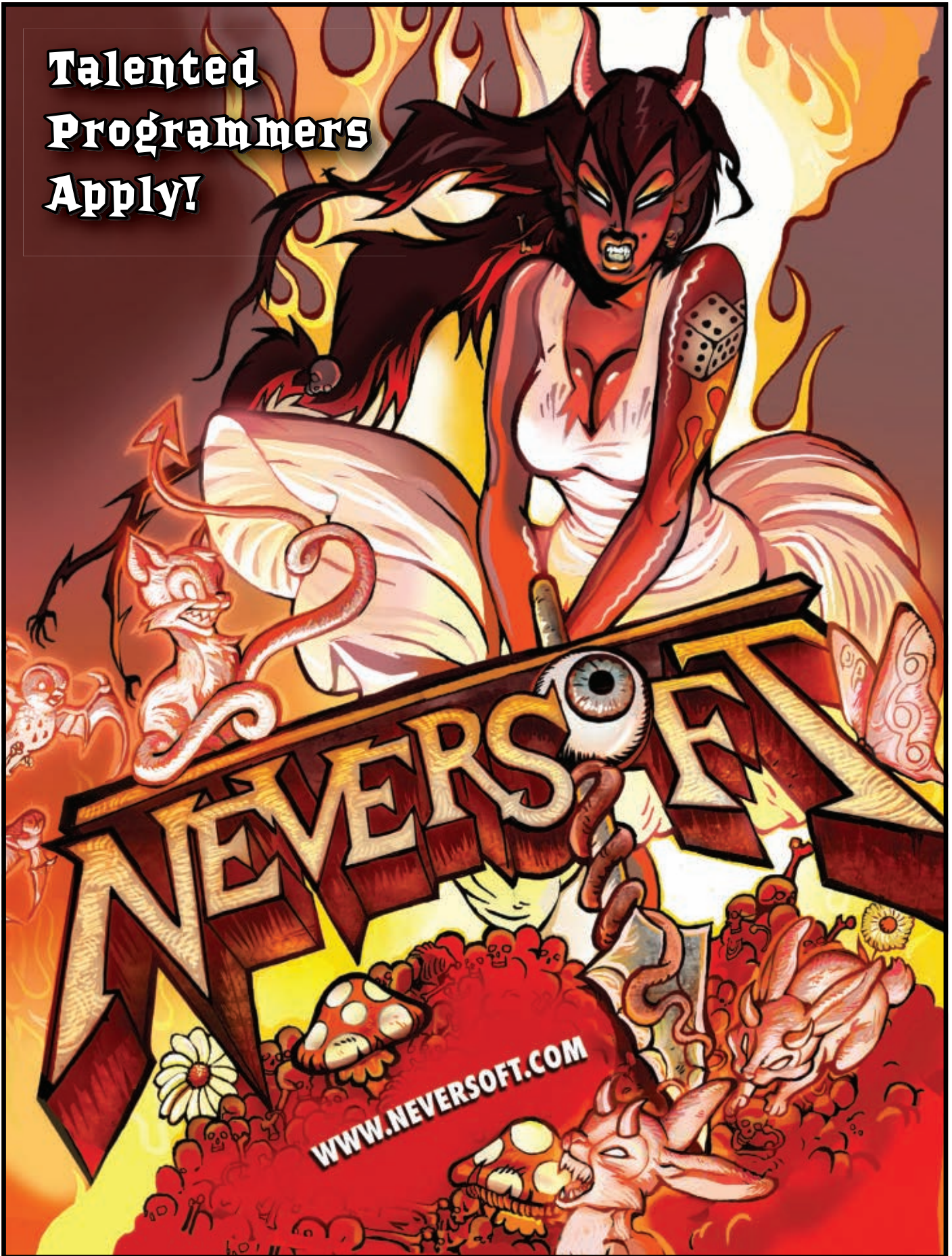
PLAYING THE ODDS

DESIGNING FOR LUCK

ONE OF THE MOST POWERFUL TOOLS a designer can use when developing games is probability—using random chance to determine the outcome of player actions or to build the environment in which play occurs. But the use of luck is not without its pitfalls, and designers should be aware of the trade-offs involved—chance can add to the experience, but it can also be counterproductive.

FAILING AT PROBABILITY

>> One challenge with using randomness is that humans are notoriously poor at accurately evaluating probability. A common example is the Gambler's Fallacy, which is the belief that odds will even out over time. If the roulette wheel comes up black five times in a row, players often believe that the odds of coming up black again are quite small, even though clearly the streak makes no difference whatsoever. Conversely, people also see streaks where none actually exist—the shooter with a "hot hand" in basketball for example. Studies show that, if anything, a successful shot actually predicts a subsequent miss.

Also, as designers of slot machines and MMOs are quite aware, setting odds unevenly between each progressive reward level makes players think that the game is more generous than it really is. One commercial slot machine had its payout odds published by wizardofodds.com in 2008:

1:1 per 8 plays
2:1 per 600 plays
5:1 per 33 plays
20:1 per 2,320 plays
80:1 per 219 plays
150:1 per 6,241 plays

The 80:1 payoff is common enough to give players the thrill of beating the odds for a big win but still rare enough that the casino is in no risk of losing money. Furthermore, humans have a hard time estimating extreme odds—a 1 percent chance is anticipated too often and 99 percent odds are considered to be as safe as 100 percent.

LEVELING THE FIELD

>> These difficulties in accurately estimating odds actually work in the favor of the game designer. Simple game design systems, such as the dice-based resource generation system in *Settlers of Catan*, can be tantalizingly difficult to master, with a dash of probability.

In fact, luck makes a game more accessible because it shrinks the gap—whether in perception or in reality—between experts and novices. In a game with a strong luck element, beginners believe that, no matter what, they have a chance to win. Few people would be willing to play a chess Grandmaster, but playing a backgammon expert is much more appealing—a few lucky throws can give anyone a chance.

In the words of M.U.L.E. designer Dani Bunten, "Although most players hate the idea of random events that will destroy their nice safe predictable strategies, nothing keeps a game alive like a wrench in the works. Do not allow players to decide this issue. They don't know it but we're offering them an excuse for when they lose ('It was that damn random event that did me in!') and an opportunity to 'beat the odds' when they win."

Thus, luck serves as a social lubricant—the alcohol of gaming, so to speak—that increases the appeal of multiplayer gaming to audiences which would not normally be suited for cutthroat head-to-head competition.

WHERE LUCK FAILS

>> Nonetheless, randomness is not appropriate for all situations or even all games. The "nasty surprise" mechanic is never a good idea. If a crate provides ammo and other bonuses when opened but explodes 1 percent of the time, the player has no chance to learn the probabilities in a safe manner. If the explosion occurs early enough, the player will immediately stop opening crates. If it happens much later, the player will feel unprepared and cheated.

Also, when randomness becomes just noise, the luck simply detracts from the player's understanding of the game. If a die roll is made every time a STARCRAFT Marine shoots at a target, the rate of fire will simply appear uneven. Over time, the effect of luck on the game's outcome will be negligible, but the player will have a harder time grasping how strong a Marine's attack actually is with all the extra random noise.

Further, luck can slow down a game unnecessarily. The board games *History of the World* and *Small World* have a very similar conquest mechanic, except that the former uses dice and the latter does not (until the final attack). Making a die roll with each attack causes a *History of the World* turn to last at least three or four times as long as a turn in *Small World*. The reason is not just the logistical issues of rolling so many dice—knowing that the results of one's decisions are completely predictable allows one to plan out all the steps at once without

AGE OF EMPIRES III

worrying about contingencies. Often, handling contingencies are a core part of the game design, but game speed is an important factor too, so designers should be sure that the trade-off is worthwhile.

Finally, luck is very inappropriate for calculations that determine victory. Unlucky rolls feel the fairest the longer players are given to react to them before the game's end. Thus, the earlier luck plays a role, the better for the perception of game balance. Many classic card games—pinochle, bridge, hearts—follow a standard model of an initial random distribution of cards that establishes the game's "terrain," followed by a luck-free series of tricks which determines the winners and losers.

PROBABILITY IS CONTENT

>> Indeed, the idea that randomness can provide an initial challenge to be overcome plays an important role in many classic games, from simple games like MINESWEEPER to deeper ones like NETHACK and AGE OF EMPIRES. At their core, solitaire and DIABLO are not so different—both present a randomly-generated environment

that the player needs to navigate intelligently for success.

An interesting recent use of randomness was SPELUNKY, which is indie developer Derek Yu's combination of the random level generation of NETHACK with the game mechanics of 2D platformers like LODE RUNNER. The addictiveness of the game comes from the unlimited number of new caverns to explore, but frustration can emerge from the wild difficulty of certain unplanned combinations of monsters and tunnels.

In fact, pure randomness can be an untamed beast, creating game dynamics that throw an otherwise solid design out of balance. For example, CIVILIZATION 3 introduced the concept of strategic resources which were required for the construction of certain units—chariots need horses, tanks need oil, and so on. These resources were sprinkled randomly across the world, which inevitably led to large continents with the only cluster of iron controlled by a single AI opponent. Complaints of being unable to field armies for lack of resources were common within the community.

For CIVILIZATION 4, the problem was solved by adding a minimum amount of space between certain important resources, so that two sources of iron could never be within seven tiles of each other. The result was a still-unpredictable arrangement of resources around the globe but without the clustering that could doom an unfortunate player. On the other hand, the game actively encouraged clustering for less important luxury resources—incense, gems, spices—to promote interesting trade dynamics.

SHOWING THE ODDS

>> Ultimately, when considering the role of probability, designers need to ask themselves: "How is luck helping or hurting the game?" Is randomness keeping the players pleasantly off-balance so they can't solve the game trivially? Or is it making the experience frustratingly unpredictable so that players are not invested in their decisions? One factor which helps ensure the former is making the probability as explicit as possible.

The strategy game ARMAGEDDON EMPIRES based combat on a few

simple die rolls and then *showed* the dice directly on-screen. Allowing the players to peer into the game's calculations increases their comfort level with the mechanics, which makes chance a tool for the player instead of a mystery.

Similarly, with CIVILIZATION 4, we introduced a help mode which showed the exact probability of success in combat, which drastically increased player satisfaction with the underlying mechanics. Because humans have such a hard time estimating probability accurately, helping them make a smart decision can improve the experience immensely.

Some deck-building card games (such as *Magic: The Gathering* or *Dominion*) put probability in the foreground by centering the game experience on the likelihood of drawing cards in the player's carefully constructed deck. These games are won by players who understand the proper ratio of rares to commons, knowing that each card will be drawn exactly once each time through the deck. This concept can be extended to other games of chance by providing, for example, a virtual "deck of dice" that ensures the distribution of die rolls is exactly even.

Another interesting—and perhaps underused—idea from the distant past of gaming history is the "Element of Chance" game option from the turn-based strategy game LORDS OF CONQUEST. The three options available—Low, Medium, and High—determined whether luck was only used to break ties or to play a larger role in resolving combat. The appropriate role of chance in a game is ultimately a subjective question, but giving players the ability to adjust the knobs themselves can open up the game to a larger audience with a greater variety of tastes. 🎮

SOREN JOHNSON *is a designer/programmer at EA Maxis, working on an unannounced project. He was the lead designer of* CIVILIZATION IV *and the co-designer of* CIVILIZATION III. *Read more of his thoughts on game design at www.designer-notes.com. Email him at* **sjohnson@gdmag.com**.

LESS IS MORE

BUILDING BOUNDARIES TO UNLEASH CREATIVITY

LucasArts' LUCIDITY.
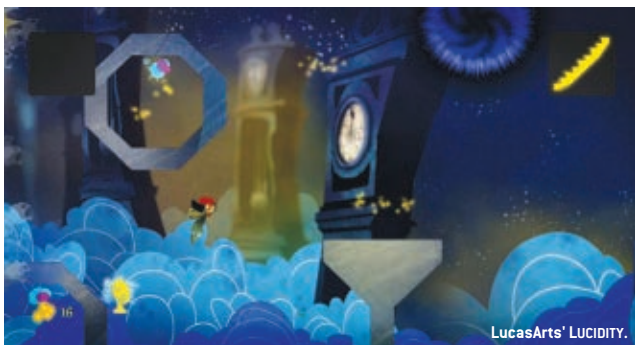
WHEN I WAS IN SCHOOL STUDYING film scoring, the composers had weekly recording sessions. Every week, 30 composers would get fifteen minutes each at the conductor's podium to record the cue they'd written for one of three film clips. In order to streamline the sessions, the teachers made us all write using the same pre-determined list of instruments. Each week the list of instruments would change, but the number always remained the same: only twelve instruments.

Limiting the orchestral palette to only twelve instruments meant that familiar crutches such as lush string lines, armies of percussionists and choirs weren't available to lean on for easy support. Instead, we had to learn how to make a bass clarinet, two cellos, an alto flute, and a cuica feel just as dramatic as if we had the London Symphony Orchestra at our disposal. The results were always fascinating cues that bucked convention, shied away from clichés, and mixed dramatic storytelling with a signature sound.

As 2009 wanes, our industry is feeling the constrictions of the global recession. Studios are closing, layoffs abound, and everyone is being asked to do more with less. Add to that the explosion of the casual and independent game markets—either via Flash games or the iPhone's App store—and it's clear that budgets are shrinking rather than growing. But shrinking budgets can be the perfect catalyst to shake off some tired game music clichés and start exploring the possibilities allowed by limiting your palette.

GOOD COMPANY

>> Placing boundaries on a composer's instrumental palette is nothing new. Bernard Herrmann's score for *Psycho* is one of the most famous film scores of all time and consists solely of an orchestral string section. The score for Jim Jarmusch's western *Dead Man* is comprised almost exclusively of nothing more than solo guitar improvisations performed by Neil Young. Thomas Newman's score to AMERICAN BEAUTY sparked a marimba fad in film, TV, and commercial scoring when it was released in 1999.

Games, unfortunately, have been slow to follow suit with limited palettes. Too often, composers simply take a limited budget to mean a turn toward vast arrays of samples rather than vast ensembles of live players. Hideki Sakamoto's captivating string quartet score to ECHOCHROME is a standout exception, as is the recent score for Sucker Punch's INFAMOUS. With INFAMOUS, Sony's music crew centered the score on the creation of junkyard instruments—everything from bungee cord plucks to windshield percussion—that were sampled and then handed to electronica artist Amon Tobin who translated the unique textures into in-game compositions. The result is a signature score that is heavy with industrial grit and exotic without being ethnic.

A CASE IN POINT

>> This past May, I had the opportunity to write the score for LucasArts' LUCIDITY. Since it's set in a 10 year-old girl's imaginary dream world, I decided the score should change from track to track with the disjointed shifts that often accompany sleep. Snippets of jazz, old foreign folk lullabies, and a hodge-podge of classical influences served as my building blocks. However, in order to reign in my budget and my time so as to fall in line with the scope of the project, I set a basic guideline at the very outset of composition: each track would contain only twelve instruments.

For each track, I changed my twelve instruments of choice. Sometimes I defined an "instrument" as its real life counterpart while other times I defined "instrument" as a particular sample patch. As such, sometimes one instrument on my list might contain five different sample patches of different articulations while other times a drum kit simply counted as one instrument. The idea was to set boundaries that were malleable to a given track's needs, rather than inflexibly penning myself in with needless limitations.

What resulted was a broad expansion of my creativity. I found myself writing for solo music box. I discovered the joys of instrument pairings that I might not had previously tried, such as low harp plucks with a saxophone quartet or doubling a melody line with toy piano and harpsichord. Time after time, I found myself habitually reaching for a new instrument track before stopping, rethinking, and then tackling the problem by searching for a new perspective in my existing sound set.

The ability to call upon countless sample instruments on a whim and a false expectation that bigger games necessitate full orchestras have both turned many game scores into bloated expanses of homogenous timbres. With the ever-present drive among development teams to create instantly identifiable gameplay mechanics and visual styles, we as composers owe it to the teams' visions to strive for an equal amount of individuality. By limiting their palettes, game composers will find a greater diversity of sound and move away from familiar habits toward the inevitable creation of signature scores. 🎵

JESSE HARLIN has been composing music for games since 1999. He is currently the staff composer for LucasArts.You can email him at jharlin@gdmag.com.

Florida Interactive Entertainment Academy

OUR MASTER'S DEGREE HAS ONE THING ON ITS MIND.

YOUR NEXT JOB. At FIEA everything from our industry-based curriculum to our new MOCAP studio is geared toward teaching you what you need to know to become a successful video game producer, programmer or artist.

Earn an accredited Master's degree in 16 months while being mentored by industry veterans who have shipped more than 40 games and films. So tip your cap to our 95% placement rate and learn more at **www.fiea.ucf.edu.**

Producer, artist, programmer track | Fully accredited Master's degree | 95% placement rate | Financial aid available | www.fiea.ucf.edu

Florida Interactive Entertainment Academy
University of Central Florida
500 West Livingston St.
Orlando, FL 32801
407-823-2121 • info@fiea.ucf.edu

UCF
CENTER FOR
EMERGING MEDIA

FIEA is a part of the University of Central Florida, a leader in modeling and simulation, optics and photonics, computer science and many other disciplines.

ADVERTISER INDEX

OUR IMPORTANT MEETING

YOUR BOSS FINALLY WORKS YOU INTO HIS SCHEDULE

HEY THERE! SORRY FOR MAKING YOU RESCHEDULE OUR MEETING SO many times. I've been so busy lately that it's a miracle I could find an open slot this whole week. You weren't waiting too long for me, were you? I was in another meeting before this and it ended up running a little late. You know how it is with folks here. Anyway, it's been such a crazy day, I seriously can't keep everything straight in my head, and to be honest I completely forgot what this one-on-one was going to be about. Could you remind me again about the agenda you sent me?

Oh, wait a second, hold on a moment—that ring tone means it's my wife. Just one second. Hello? Yeah. Yeah. Okay, honey. Okay. Look, uh, I'm in a meeting right now. A meeting. Yeah. I'll call you back. I'll call you back. I'll … okay. Well in that case, just get those frozen chicken burritos. No, I'm not worried it seems like we're giving up. There's nothing wrong with having some frozen food around. For convenience's sake. No, I thought they tasted fine. I think she'll be fine with it. What could she—how could she possibly complain? They're the same as—listen, I gotta go. Yeah. Sorry. Yeah. Sorry. Okay. Love you. Bye. Okay, I will. Bye. All right. Sorry about that. Anyway, where were we? Oh, that's right, the meeting agenda. Don't worry about digging it up, I can find it right here in Outlook. Uh … where is it … you sent it just yesterday, right? I can't find this e-mail. Let me try sorting by your name. No, that's not working. Crap. I'll do a quick search. Yeah. Man, why does Outlook search take so long? How can Google search the whole Internet instantly but Outlook takes an hour to search my inbox? Let me try sorting by date. What's today's date again? It's so strange, I'm just not seeing it. I totally remember you sending it. Oh! I was looking at last year. Okay, I think I'm in the right year now—ha! Oh, man, that was funny! Remember when Darren sent this picture out to the whole company? I laughed for ages. That was great. I love Darren's picture emails. Like the one that had Paul's head pasted onto a woman's body. Hilarious. Or the one with Zach Photoshopped into a dinosaur. You haven't seen that one? Are you serious? Oh man, you have to see it, you've got to. Hold on, let me look it up. I saved it into my "funny pictures" folder. It's in here somewhere. Ha, here it is! Oh, man, that makes me laugh. Isn't that great? Darren really is the master of Photoshop. Even Caleb can't touch him.

So. Time to have our meeting for real now. Go ahead and tell me what it is you wanted to discuss? Uh-huh. Uh-huh. Keep talking, I'm totally listening—I'm just going to respond to this IM really quick while you're speaking. Uh-huh. Uh-huh. Hey, have you seen this? This new IM client I got has a bunch of new emoticons in it. Check it out. Look, that one is drinking beer. Isn't that funny? It's like he's going "glug, glug." That one comes in handy all the time. Here's another funny one. The little antlers are hilarious. Look at those! I love these. Here, let me e-mail you the URL where you can find this program. You should download it, so I can send you messages with beer icons in them. Wouldn't that be awesome? Anyway, go on …

Who's that at the door? Is it Jim? Hey, bro! Come on in … well, we're kind of having a meeting, but if it's quick, that's fine. What's up? What? Are you kidding? No. That can't be right. Are you sure about this? Well, I'll sort it out with him then. Yeah. Thanks for letting me know. What? Oh yeah, lunch tomorrow sounds great. See you then. By the way, did you get that e-mail I sent you? I sent it like fifteen minutes ago. Well, when you get back to your desk, check it out, let me know what you think. You can read it when you get back, but basically, I was trying to play foosball with Chris the other day, and man, that table is seriously janky! I know that's Bob's old table and he knows all its weird little secrets and everything, but for fair break room competition we ought to buy a brand new one, don't you think? Anyway, it's all in the e-mail. I found some pretty cool foosball tables online. Like there's one with a custom laminate on it, called the Meteor Fireball or something. I know! It looked real sweet. Yeah. Hey look, I, uh, I gotta get back to this meeting here. Yeah, talk to you later. Remember—Meteor Fireball! Something like that. Dude, it would be so awesome! Alright bro.

Whew! Boy, can't get any quiet around here, huh? I apologize for the interruptions. Let's continue where we left off. Uh-huh. Mmm. Wait, what did you say just now? I was checking for new Tweets on my iPhone and missed that last bit.

Oh, you want another meeting? Well, I'm seriously slammed right now … do you think it could wait a bit? I'm not trying to blow you off, seriously. It's just total crazy time at the moment. I mean I certainly can see why you have the concerns you voiced to me just now, but I think we'll be fine if we can just, you know, hunker down, and stay focused. 🎮

Autodesk Games Insight

The latest scoop from Autodesk Media & Entertainment

Image courtesy of Mixamo

> "The huge advantage of HumanIK is that it's extremely fast and optimized from a code perspective. We're running everything in real-time."
>
> — Stefano Corazza
> Chief Technology Officer
> Mixamo

Mixamo combines cutting-edge research and Autodesk games technology to help accelerate the character animation process

Mixamo is an online animation service that helps users to create production-quality 3D character animations on the web, that are customized and retargeted to their own characters. The Mixamo website (www.mixamo.com) offers a large collection of motion models, which can be customized for many game studio needs using an intuitive interface powered by advanced technology underneath.

"Mixamo is geared toward game studios who face project constraints, whether that's related to reduced budget or production cycle deadlines," says Jon Burns, Product Marketing Manager for Mixamo. "The coolest thing we offer is the initial baseline lifting for game artists, which frees them up to focus on the creative aspect of their character animation. It fits quite well with iPhone and other mobile/handheld game developers who don't have much time, yet want high quality productions."

The underlying technology of Mixamo stems from 4 years of research by Stefano Corazza, CTO, at the Stanford Biomotion Lab. His research was focused on two concepts: markerless motion capture and generative models. Using generative models helps give Mixamo users the ability to customize motions into hundreds of thousands

of unique combinations using intuitive sliders. Mixamo also uses markerless motion capture technology which enables difficult motion captures, such as quadruped animals, which will be rolled out on the service soon.

To implement Mixamo, the team relied on several key Autodesk technologies. Autodesk® MotionBuilder® software is used extensively for the motion capture of the clips, Autodesk® HumanIK® animation middleware is integrated into the website service to provide real-time IK adjustments and Autodesk® FBX® was chosen as the format to manage and move data.

"Our mocap pipeline relies heavily on MotionBuilder — which we love," states Stefano. "It's very important that what we see on the web is the same as what we see in MotionBuilder. That's why we use HumanIK. The huge advantage of HumanIK is that it's extremely fast and optimized from a code perspective. We're running everything in real-time."

"Other services have approaches where you tweak a model or animation, and it takes several seconds or minutes to update the result. Mixamo animation customization is all real-time. We needed something that responds very quickly, which HumanIK provides. Since we're using FBX and the fact that HumanIK, MotionBuilder and FBX are integrated, there was added value in using all three technologies collectively."

Why did such a capable team decide to integrate HumanIK instead of writing their own animation middleware? "From the moment we got HumanIK to the moment we were first able to run it on our pipeline was approximately one week. This is very fast in terms of integration. The product is very modular and can be plugged easily into existing pipelines."

"If you were to write your own IK system, you would develop for 6 months to 1 year and wouldn't really be sure if it met the quality bar until that initial, heavy investment," continues Stefano. "This is a huge unknown especially for a start up. It was a risk management decision. Because HumanIK is integrated into MotionBuilder, we were able to test it before buying the middleware. We knew that HumanIK would deliver the same thing. This was a good way of testing the product before we integrated it."
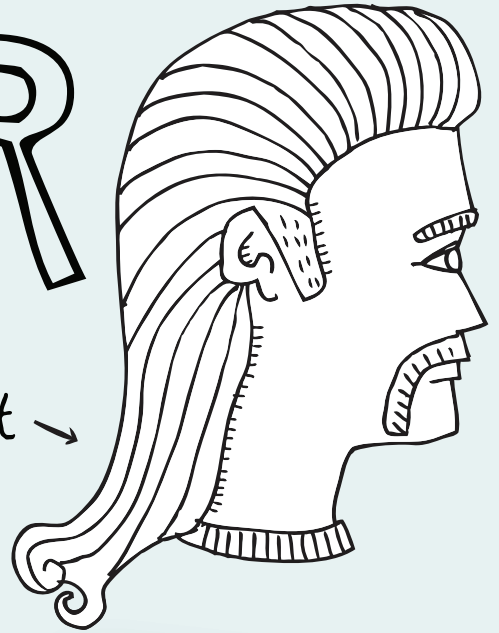
Mixamo demonstrates the merging of several impressive technologies: generative models and markerless motion capture with Autodesk HumanIK, MotionBuilder and FBX. This combination improves the 3D character animation process, helping to reduce production times and boost the creative output of 3D game artists and design professionals.

For more information on Autodesk games software and middleware please visit **www.autodesk.com/games**.

Autodesk®